

AD-A237 125



①

ADVANCED DECISION SYSTEMS

1500 Plymouth Street, Mountain View, California 94043-1230 • 415-960-7300 • FAX (415) 960-7500

ADS TR-1128-02

88 - 0 / 83

DISTRIBUTED TRACKING IN  
DISTRIBUTED SENSOR NETWORKS

May 26, 1988

FINAL REPORT

Contract No. MDA903-86-C-0011  
Contract Expiration Date: May 26, 1988  
Total Dollar Value: \$722,192  
Government Sponsor: DARPA/ISTO  
DARPA Order No. 4272/07

OTIC  
S  
J

Advanced Decision Systems

C.Y. Chong  
K.C. Chang  
S. Mori  
D.S. Spain

Qualcomm, Inc.  
J.K. Wolf (et al.)

DEFINITION OF DOCUMENT A  
Approved for public release  
Distribution unlimited

Prepared for:

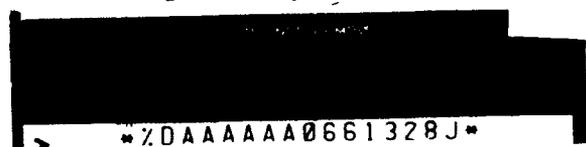
Defense Advanced Research  
Projects Agency/ISTO  
1400 Wilson Boulevard  
Arlington, VA 22209

and

Defense Supply Service - Washington  
Washington, DC 20310

The views, opinions, and findings contained in this report are those of the authors and should not be construed as an official Department of Defense position, policy, or decision, unless so designated by other official documentation.

91 6 17 049



\*ZDAAAAAA0661328J\*

AD-A237 125

91-02352



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

## REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED		1b RESTRICTIVE MARKINGS	
2a SECURITY CLASSIFICATION AUTHORITY N/A		3 DISTRIBUTION/AVAILABILITY OF REPORT  Distribution Unlimited	
2b DECLASSIFICATION/DOWNGRADING SCHEDULE N/A		5 MONITORING ORGANIZATION REPORT NUMBER(S) UNLIMITED	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)  ADS TR-1128-02		7a NAME OF MONITORING ORGANIZATION  Defense Supply Service - Washington	
6a NAME OF PERFORMING ORGANIZATION  Advanced Decision Systems	6b OFFICE SYMBOL (if applicable)	7b ADDRESS (City, State, and ZIP Code)  Room 1D-245, The Pentagon Washington, DC 20310	
6c ADDRESS (City, State, and ZIP Code)  1500 Plymouth Street Mountain View, CA 94043		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
9a. NAME OF FUNDING / SPONSORING ORGANIZATION Defense Advanced Research Projects Agency	8b. OFFICE SYMBOL (if applicable) ISTO	10 SOURCE OF FUNDING NUMBERS	
8c. ADDRESS (City, State, and ZIP Code)  1400 Wilson Boulevard Arlington, VA 22209		PROGRAM ELEMENT NO.	PROJECT NO.
		TASK NO.	WORK UNIT ACCESSION NO.
		ARPA Order No. 4272/07	

11 TITLE (Include Security Classification)

DISTRIBUTED TRACKING IN DISTRIBUTED SENSOR NETWORKS

12 PERSONAL AUTHOR(S)

C.Y. Chong, K.C. Chang, S. Mori, D.S. Spain

13a. TYPE OF REPORT

Final Technical

13b. TIME COVERED

FROM 86/02/27 TO 88/05/26

14. DATE OF REPORT (Year, Month, Day)

1988 May 26

15 PAGE COUNT

269

16 SUPPLEMENTARY NOTATION

17 COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

Distributed Tracking, Distributed Estimation, Distributed Sensor, Acoustic Tracking, Passive Tracking, Sensor Fusion, Hypothesis Processing, Multitarget Tracking

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

This report documents research results on distributed tracking by a distributed sensor network (DSN). A DSN is made up of a set of nodes which can communicate to each other via a communication network. Each DSN node contains a processor collecting data from some sensors. The processor performs tracking functions using the local sensor data and communicates the processing results to other nodes according to some communication strategy. The receiving node then integrates or fuses the information from other nodes with its local information to arrive at a better estimate. Compared with a centralized tracking system, a DSN has advantages such as increased reliability, less communication, local use of processing results, etc.

(OVER)

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT

 UNCLASSIFIED/UNLIMITED
  SAME AS RPT
  DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION

UNCLASSIFIED

22a NAME OF RESPONSIBLE INDIVIDUAL

22b TELEPHONE (Include Area Code)

22c. OFFICE SYMBOL

SECURITY CLASSIFICATION OF THIS PAGE

Under previous DSN contracts, a general theory for distributed tracking based on the multiple-hypothesis approach had been developed for difficult environments involving many targets, high false alarm rates, poor detection conditions, etc. The objective of current project was to apply this theory to the tracking of air targets by a network of acoustic sensors.

Because of acoustic sensor characteristics such as large propagation delay relative to target speed, azimuth only measurements and poor sensor resolution, the general algorithm had to be modified. In the modified system, tracks are classified into local or global according to the target state distribution. Local tracks have azimuths and sound pressures (and rates) as states and are formed from a single node before any communication. Global tracks have positions and velocities as states and are initiated when two nodes communicate. The algorithm also accounts for merged measurement from targets which are close to each other. The information distribution strategy is adaptive and communicates only when the information will be useful to another node.

The algorithms have been implemented on a Symbolics LISP Machine. Simulations have been performed using synthetic data for scenarios involving different target and sensor configurations, as well as pre-recorded real data. The algorithms have been found to perform satisfactorily except for targets that are too close together to be resolved by the sensors. The feasibility of performing distributed situation assessment by a network of autonomous but cooperating agents was thus demonstrated.

An appendix contains the results of investigating tracking from a communication or information-theoretic view point. This work was performed by Qualcomm, Inc. under a subcontract.

# DISTRIBUTED TRACKING IN DISTRIBUTED SENSOR NETWORKS

## FINAL REPORT

Accession For:	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution / _____	
Availability Codes	
Dist	Availability / or Special
A-1	



Prepared for:  
 DARPA  
 1400 Wilson Boulevard  
 Arlington, VA 22209

and

Defense Supply Service -Washington  
 Washington, DC 20310

Prepared by:  
 Advanced Decision Systems  
 C.Y. Chong  
 K.C. Chang  
 S. Mori  
 D.S. Spain

Qualcomm, Inc.  
 J.K. Wolf (et al.)

Approved for public release: distribution unlimited.

## ABSTRACT

This report documents research results on distributed tracking by a distributed sensor network (DSN). A DSN is made up of a set of nodes which can communicate to each other via a communication network. Each DSN node contains a processor collecting data from some sensors. The processor performs tracking functions using the local sensor data and communicates the processing results to other nodes according to some communication strategy. The receiving node then integrates or fuses the information from other nodes with its local information to arrive at a better estimate. Compared with a centralized tracking system, a DSN has advantages such as increased reliability, less communication, local use of processing results, etc.

Under previous DSN contracts, a general theory for distributed tracking based on the multiple-hypothesis approach had been developed for difficult environments involving many targets, high false alarm rates, poor detection conditions, etc. The objective of current project was to apply this theory to the tracking of air targets by a network of acoustic sensors.

Because of acoustic sensor characteristics such as large propagation delay relative to target speed, azimuth only measurements and poor sensor resolution, the general algorithm had to be modified. In the modified system, tracks are classified into local or global according to the target state distribution. Local tracks have azimuths and sound pressures (and rates) as states and are formed from a single node before any communication. Global tracks have positions and velocities as states and are initiated when two nodes communicate. The algorithm also accounts for merged measurements from targets which are close to each other. The information distribution strategy is adaptive and communicates only when the information will be useful to another node.

The algorithms have been implemented on a Symbolics LISP Machine. Simulations have been performed using synthetic data for scenarios involving different target and sensor configurations, as well as pre-recorded real data. The algorithms have been found to perform satisfactorily except for targets that are too close together to be resolved by the sensors. The feasibility of performing distributed situation assessment by a network of autonomous but cooperating agents was thus demonstrated.

An appendix contains the results of investigating tracking from a communication or information-theoretic view point. This work was performed by Qualcomm, Inc. under a subcontract.

## NOTE TO READER

This document is the Advanced Decision System (ADS) Final Technical report on the Distributed Tracking in Distributed Sensor Networks. This volume contains the results of efforts undertaken by Qualcomm, Inc. working as subcontractor to ADS on the effort. Qualcomm's Final Report entitled, "Distributed Sensor Program," is attached in its entirety as Appendix C entitled, "Tracking from Communication and Information Theoretic View Point."

## TABLE OF CONTENTS

	Page
1. INTRODUCTION AND SUMMARY	1-1
1.1 GENERAL DSN APPROACH	1-1
1.1.1 Generalized Tracker/Classifier	1-3
1.1.2 Information Fusion	1-3
1.1.3 Information Distribution	1-6
1.1.4 Resource allocation	1-6
1.2 PROJECT GOALS	1-8
1.3 PROJECT ACCOMPLISHMENTS	1-9
1.4 REPORT ORGANIZATION	1-11
2. DISTRIBUTED MULTITARGET TRACKING	2-1
2.1 NODAL STRUCTURE	2-1
2.1.1 Local tracking data	2-1
2.1.2 Processing structure	2-4
2.2 LOCAL HYPOTHESIS PROCESSING	2-6
2.2.1 Hypothesis formation	2-6
2.2.2 Hypothesis Evaluation	2-12
2.2.3 Hypothesis management	2-15
2.3 HYPOTHESIS PROCESSING IN FUSION	2-16
2.3.1 Fusability and information graph	2-16
2.3.2 Hypothesis formation	2-21
2.3.3 Hypothesis evaluation	2-23
2.3.4 Hypothesis management	2-24
2.4 CONSTRUCTION OF INFORMATION GRAPH	2-24
2.4.1 Limited range broadcast communication	2-25
2.4.2 Lost messages and communication failures	2-28
3. DISTRIBUTED ACOUSTIC SYSTEM AND MODELS	3-1
3.1 SYSTEM DESCRIPTION	3-1
3.2 MODELS	3-4
4. LOCAL DATA PROCESSING	4-1
4.1 TRACK REPRESENTATION	4-1
4.2 HYPOTHESIS FORMATION	4-4
4.3 HYPOTHESIS EVALUATION	4-4

4.4 TRACK UPDATE	4-10
4.5 MULTIPLE MODEL APPROACH	4-12
5. INFORMATION FUSION AND DISTRIBUTION	5-1
5.1 FUSION PROCESSING	5-1
5.1.1 Hypothesis Formation	5-2
5.1.2 Hypothesis Evaluation	5-2
5.1.3 Hypothesis Management	5-11
5.2 INFORMATION DISTRIBUTION	5-11
5.2.1 Communication Criteria	5-12
5.2.2 Communication Pattern and Information Graph	5-13
6. SIMULATION RESULTS	6-1
6.1 SIMULATION ENVIRONMENT	6-1
6.1.1 Data Generator	6-1
6.1.2 User Interface	6-1
6.1.3 Simulation Controller	6-8
6.1.4 Communication Controller	6-9
6.2 ALGORITHM FEATURES	6-11
6.2.1 Use of Sound Pressure Track	6-12
6.2.2 Global Track Initiation	6-13
6.2.3 Multiple Model Approach	6-13
6.2.4 Merged Measurement Models	6-21
6.2.5 Selection of Communication Strategies	6-21
6.3 SIMULATION RESULTS	6-21
6.3.1 Simulations with Synthetic Data	6-21
6.3.2 Simulation with Real Data	6-48
6.4 MONTE CARLO RESULTS	6-57
6.4.1 Scenario I (Single Constant Speed Target)	6-64
6.4.2 Scenario II (Single Maneuvering Target)	6-65
6.4.3 Scenario III (Two Targets in Line Formation)	6-65
7. CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH	7-1
7.1 CONCLUSIONS	7-1
7.2 SUGGESTIONS FOR FUTURE RESEARCH	7-2
8. REFERENCES	8-1
APPENDIX A. MERGED MEASUREMENT LIKELIHOOD CALCULATION	A-1
APPENDIX B. GEOMETRY IN 2-D SUBSONIC ACOUSTIC TRACKING	B-1
APPENDIX C. TRACKING FROM COMMUNICATION AND INFORMATION THEORETIC VIEW POINT	C-1

## LIST OF FIGURES

	Page
1-1: DSN Problem	1-2
1-2: Structure of DSN Node	1-4
1-3: Generalized Tracker/Classifier	1-5
1-4: Information Fusion	1-7
2-1: Functional Structure of Each Node	2-2
2-2: Definition of Hypothesis	2-3
2-3: Example of Hypothesis Structure	2-5
2-4: Hypothesis Tree (no merged measurements)	2-10
2-5: Hypothesis Tree (with merged measurements)	2-13
2-6: Fusability of Tracks	2-17
2-7: Fusability of Hypotheses	2-18
2-8: Broadcast Communication	2-19
2-9: Cyclic Communication	2-20
2-10: Six Node Configuration	2-25
2-11: Information Graph for Sequential Broadcast	2-26
2-12: Partial Information Graphs	2-27
3-1: DSN Deployment Patterns	3-3
3-2: Target Sensor Geometry	3-5
4-1: Track Representation	4-3
4-2: Hypothesis Expansion with Measurement Merging Possibility (1)	4-5
4-3: Hypothesis Expansion with Measurement Merging Possibility (2)	4-6
5-1: Possible Track-to-Track Combinations	5-4
5-2: Comparison of Two Simple Communication Patterns	5-15
5-3: Three-Node Cyclic Communication Patterns	5-17
6-1: Simulation Environment	6-2
6-2: Scenario Window	6-4
6-3: Local Track Windows	6-5
6-4: Global Track Window	6-6
6-5: Hypothesis and Track Trees	6-7
6-6: Information Flow Graph	6-8
6-7: Agenda Queue Based Simulation Control	6-10
6-8: Scenario Window	6-14
6-9: Local Tracking Results with $Q=1.0e-8$	6-15
6-10: Local Tracking Results with $Q=1.0e-6$	6-16
6-11: Case I : Initial Glocal Track	6-17
6-12: Case II : Initial Glocal Track	6-18
6-13: Local Tracking Results with Multiple Model Approach	6-19
6-14: Model Probability History	6-20
6-15: Single-Target Scenarios	6-23
6-16: Tracking Results with Speed = 0.2 Mach (Scenario 1-1)	6-25

6-17: Tracking Results with Speed = 0.4 Mach (Scenario 1-2)	6-26
6-18: Tracking Results with Speed = 0.6 Mach (Scenario 1-3)	6-27
6-19: Tracking Results with Speed = 0.2 Mach (Scenario 1-4)	6-29
6-20: Tracking Results with Speed = 0.4 Mach (Scenario 1-5)	6-30
6-21: Tracking Results with Speed = 0.6 Mach (Scenario 1-6)	6-31
6-22: Tracking Results with speed from 0.2 to 0.6 Mach (Scenario 1-7)	6-33
6-23: Tracking Results with 15 Degree Direction Change (Scenario 1-8)	6-34
6-24: Tracking Results with 30 Degree Direction Change (Scenario 1-9)	6-35
6-25: Tracking Results with 45 Degree Direction Change (Scenario 1-10)	6-36
6-26: Two-Target Scenarios	6-39
6-27: Tracking Results of Line-Formation I (Scenario 2-1)	6-40
6-28: Tracking Results of Line-Formation II (Scenario 2-2)	6-41
6-29: Tracking Results of Parallel Formation I (Scenario 2-3)	6-43
6-30: Tracking Results of Parallel Formation II (Scenario 2-4)	6-44
6-31: Tracking Results of Parallel Formation III (Scenario 2-5)	6-45
6-32: Tracking Results of Crossing Targets I (Scenario 2-6)	6-47
6-33: Tracking Results of Crossing Targets II (Scenario 2-7)	6-49
6-34: Three-Target Scenarios	6-50
6-35: Tracking Results of Three-Target Case I (Scenario 2-8)	6-51
6-36: Tracking Results of Three-Target Case II (Scenario 2-9)	6-52
6-37: Real Data Scenario (Scenario 2-10)	6-54
6-38: Raw Data from Sensor 1	6-55
6-39: Raw Data from Sensor 2	6-56
6-40: Filtered Data from Sensor 1 - Fixed Threshold	6-58
6-41: Filtered Data from Sensor 2 - Fixed Threshold	6-59
6-42: Filtered Data from Sensor 1 - Adaptive Threshold	6-60
6-43: Filtered Data from Sensor 2 - Adaptive Threshold	6-61
6-44: Tracking Results of Real Data I	6-62
6-45: Tracking Results of Real Data II	6-63
6-46: Typical Performance Results for Case I	6-66
6-47: One-Target Constant Speed Case with False Rate = 0.5	6-67
6-48: One-Target Constant Speed Case with Comm. Period = 5	6-68
6-49: Typical Performance Results for Case II	6-69
6-50: One Maneuvering Target Case with Sensor Resolution = 10 Degrees	6-70
6-51: Typical Performance Results for Case III	6-71
6-52: Two-Target In-Line Formation Case with Sensor Resolution 10 Degrees	6-72
6-53: Two-Target In-Line Formation Case with Target Separation 1.0 KM	6-73

## LIST OF TABLES

	Page
2-1: Track-Measurement Cross Reference Table	2-8
2-2: Hypothesis Set (no merged measurements)	2-9
2-3: Extended Track-Measurement Cross Reference Table	2-11
2-4: Hypothesis Set (with merged measurements)	2-12
2-5: Track-Measurement Cross Reference Table	2-15
2-6: Track-to-Track Fusability Table	2-22
6-1: Parameters Used in Synthetic Data Generator	6-22
6-2: Performance Results - One-Target (I)	6-28
6-3: Performance Results - One Target (II)	6-32
6-4: Performance Results - One Target (III)	6-37
6-5: Performance Results - Two-Target (I)	6-42
6-6: Performance Results - Two-Targets (II)	6-46
6-7: Performance Results - Two-Target (III)	6-49
6-8: Performance Results - Three-Target Case	6-53
6-9: Performance Results of Real Data Cases	6-64

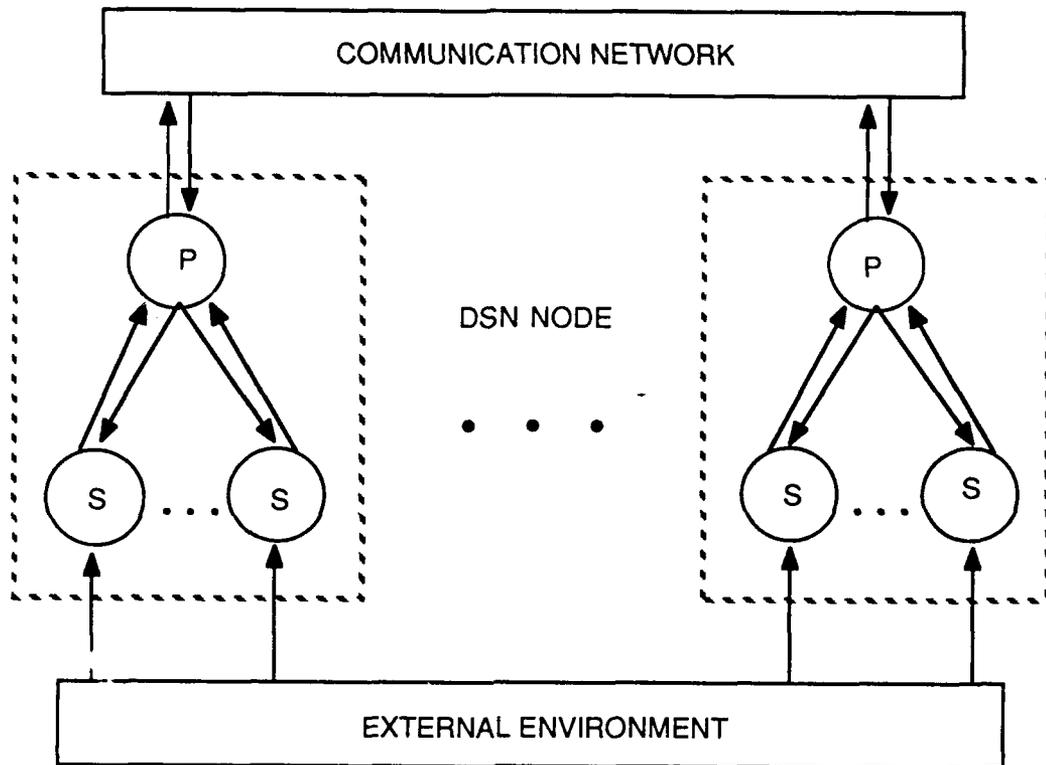
## 1. INTRODUCTION AND SUMMARY

This technical report describes the results of research performed under the contract entitled "Distributed Tracking in Distributed Sensor Networks". The major part of the research was performed at Advanced Decision Systems (ADS) and dealt with the distributed processing of acoustic sensor data for tracking air targets in a distributed sensor network (DSN). Another part of the research, concerned with the formulation of the tracking problem from a communication theorist's point of view, was performed by Qualcomm, Inc. under a subcontract.

### 1.1 GENERAL DSN APPROACH

A general DSN has the structure shown in Figure 1-1. There is a system of distributed sensor/processor nodes. Each node may have one or more sensor types, and the sensors from different nodes may have overlapping coverage. The sensors collect data from the environment and pass them on to the processors (processing nodes). The processing nodes process the sensor data and communicate with the other nodes through the communication network to obtain an assessment of the state of the world. It is generally assumed that no single node possesses complete information, and each node may have a different world model. In general, the processing nodes can also control the sensors to improve the performance of the overall system.

A DSN can be used for many applications. In our past and current work, we have been particularly interested in DSNs used for the tracking and classification of multiple targets. The target environment is assumed to be dense, so that determining the origins of the measurements in a particular sensor report is not obvious. The problem is further complicated by the presence of both false alarms and missing target reports. In such a network, tracking and classification is highly dependent on identifying the correct data association. Since the nodes in general have access to different information, communication among the nodes can improve the performance of the system. In our work [1, 2, 3] thus far, we have developed distributed processing systems for tracking and classifying multiple targets under general assumptions on target and sensor models and communication patterns.



S = SENSOR  
 P = PROCESSOR

Figure 1-1: DSN Problem

A multiple hypothesis approach has been used to solve the general problem of distributed target tracking and classification. Each node in the DSN is assumed to have the structure shown in Figure 1-2. It contains the following four modules.

### **1.1.1 Generalized Tracker/Classifier**

This module is responsible for processing the local data before any communication with the other nodes takes place. Since the objective of the system under consideration is the tracking and classification of multiple targets, this module is a multitarget tracker. In the previous projects, we have developed a general theory for multitarget tracking which is implemented in the form of the *Generalized Tracker/Classifier* (GTC). The GTC has the structure shown in Figure 1-3 and itself consists of four modules. The *hypothesis formation* module forms multiple hypotheses from the sensor data, each consisting of a collection of tracks to explain the origins of the measurements in each data set. These hypotheses are then evaluated by the *hypothesis evaluation* module with respect to their probabilities of being true. The *filtering and parameter estimation* module generates state estimates and classifications for each track. It is essential for hypothesis evaluation and can thus be viewed as a submodule. To stay within the computational constraints of each node, the hypotheses are pruned, combined, clustered, etc. This takes place in the *hypothesis management* module. The result of this processing is a set of hypotheses, and their probabilities, a collection of tracks corresponding to possible targets and the state distributions of these tracks. These quantities together constitute the information state for multitarget tracking.

### **1.1.2 Information Fusion**

This module combines the local information with information obtained from the other nodes to obtain a new situation assessment. The information from the local nodes consists of the information described above. The information from other nodes is also similar. Information fusion then consists of the following steps (see Figure 1-4):

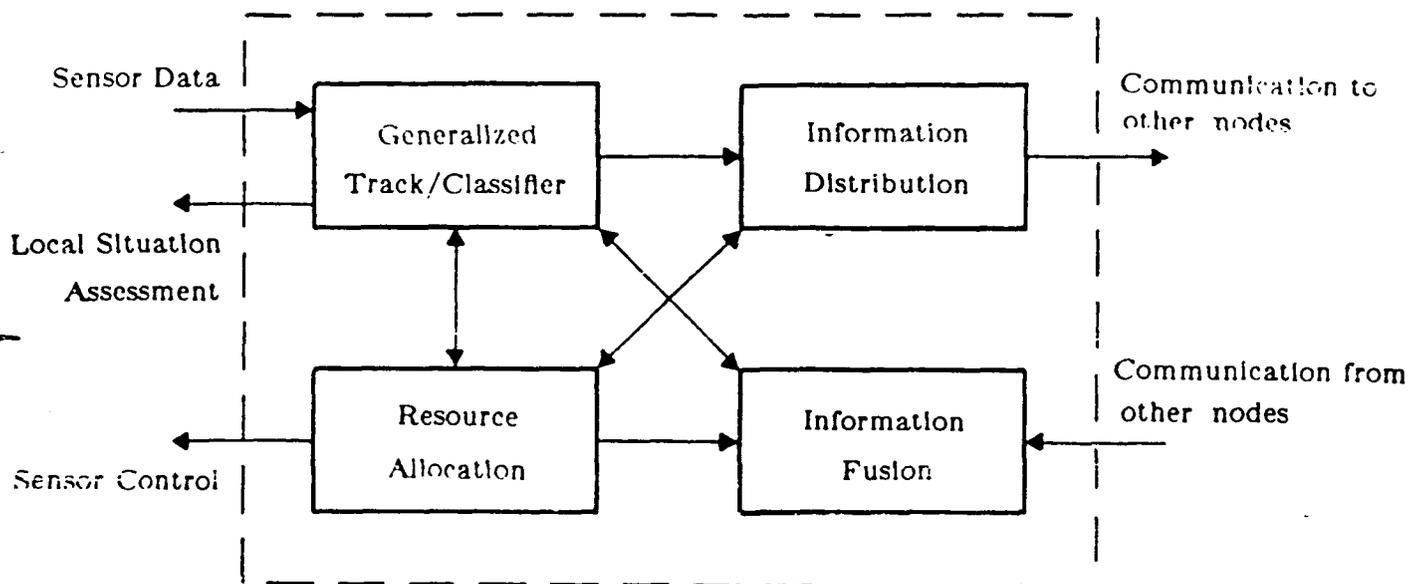


Figure 1-2: Structure of DSN Node

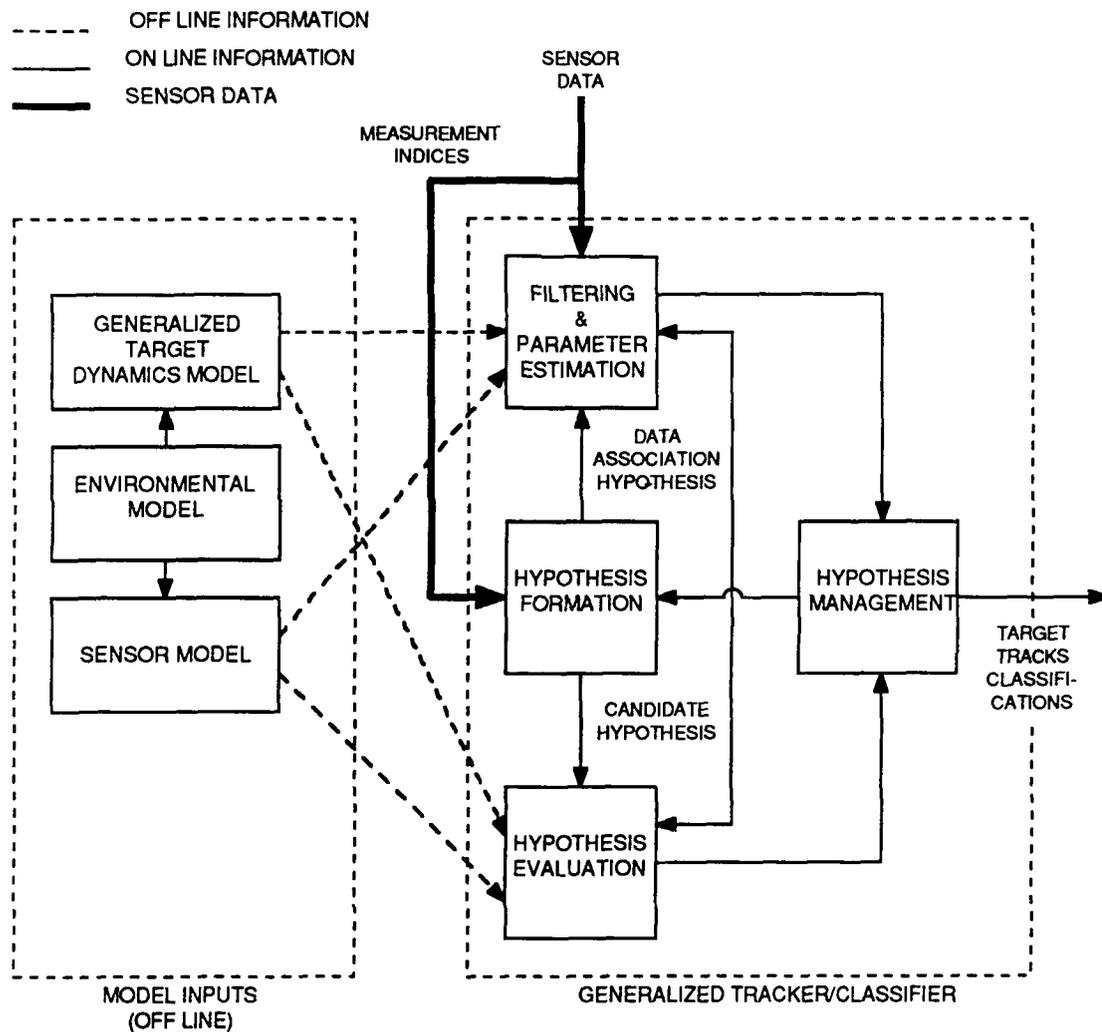


Figure 1-3: Generalized Tracker/Classifier

1. *Hypothesis Formation* - Given a set of hypotheses from other nodes, this submodule generates new global hypotheses. Tracks from the hypotheses of different nodes are associated in all possible ways, according to whether they correspond to the same or different targets.
2. *Hypothesis Evaluation* - Each of the hypotheses formed above is then evaluated with respect to its probability of being true. The statistics of the tracks from different hypotheses are used in this evaluation. For example, if two tracks are widely apart in their position or velocity distributions, they are more likely to have come from different targets than the same target.
3. *Hypothesis Management* - This is again needed to make computation feasible given the available resources.

### **1.1.3 Information Distribution**

This module decides what information is to be transmitted, who gets the information, and when it should be communicated. It thus specifies the information available to each node at any time, i.e., the information structure of the system. Information distribution can be fixed a priori for simple systems, or it can be highly adaptive to the information needs in the system.

### **1.1.4 Resource allocation**

This module allocates the resources under the control of the processing node to maintain or improve the performance of the system. Some typical resources include sensor resources and processing resources. Both resource allocation and information distribution can affect the information available in the network. Thus their activities should be coordinated.

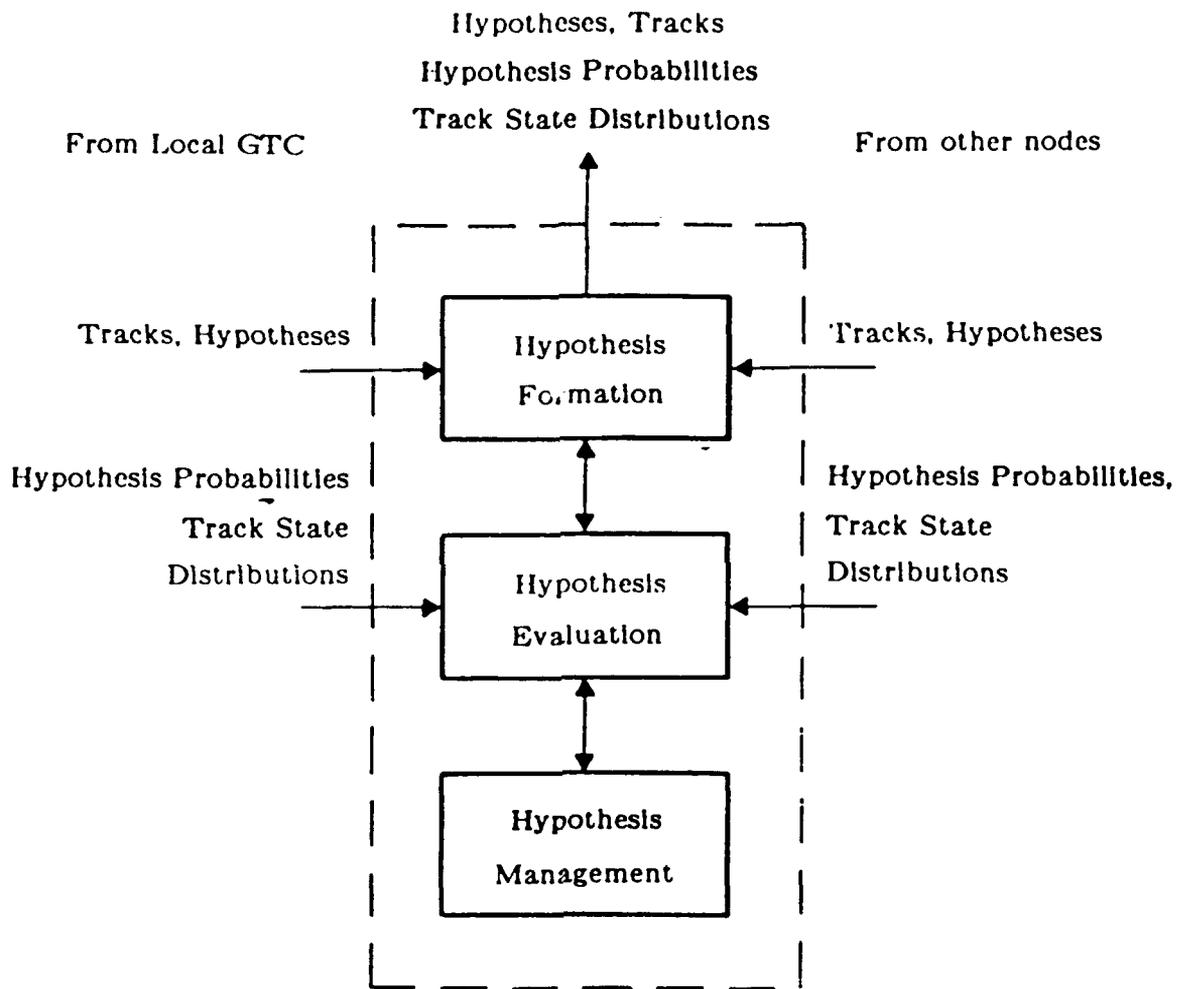


Figure 1-4: Information Fusion

## 1.2 PROJECT GOALS

As part of the DARPA DSN program, M.I.T. Lincoln Lab. performed research on the tracking of low flying aircraft using acoustic sensors. A DSN testbed was developed and used to test and demonstrate DSN techniques and technology. Their efforts concentrated on the practical aspects of creating and demonstrating the testbed. ADS' focus was on the theoretical development of algorithms for difficult environments involving many targets, high false alarm rates, poor detection conditions, etc. The goal of this project was to adapt the multiple-hypothesis approach of distributed tracking to the acoustic sensor scenario used by Lincoln Lab. An additional goal was to examine tracking from a communication theorist's perspective. Specifically, we have the following objectives:

- Designing the functional architecture for each node in the DSN
- Designing and implementing a simulation environment for the Lincoln Laboratory testbed
- Developing algorithms for tracking multiple targets using acoustic sensors in a DSN
- Implementing the algorithms in a simulated DSN
- Testing, evaluating, and demonstrating the algorithms using synthetic or pre-recorded real data

Although the general methodology of Section 1.1 was in theory applicable to the acoustic tracking problem, the specific acoustic tracking scenario raised technical issues which had to be addressed before algorithms could be developed to perform satisfactorily. Some of these issues are:

- *Azimuth-only measurements.* From a single node, the target location is not observable from the azimuth measurements. From a pair of nodes, however, a target becomes observable. An important question is thus the types of processing to be performed locally by one node and jointly by a pair of nodes. One possibility is to use different representations such as azimuth tracks for local processing and position tracks after fusion. Another possibility is to adopt a single representation.
- *Propagation delay.* Acoustic signals generated by a target do not reach a node instantaneously. Since the target speed is substantial compared to the speed of sound in the air, the delay has to be considered explicitly in any information processing. For example, the true bearing of a target at a node can be quite different from its apparent bearing at the node.
- *Poor sensor resolution.* Due to the poor sensor resolution (20 degrees separation needed before two targets can be resolved), two targets which are close together may be detected as a single target. The general algorithm developed on previous contracts largely ignored this possibility. New techniques had to be developed to handle this situation.
- *Range dependent detection.* Since target detection depends on the range, and range affects the sound pressure received at a node, some useful information may be present in the sound pressure. On the other hand, the acoustic propagation characteristics in air may be too complicated and unreliable. Whether the intensity information can be exploited or not had to be investigated.

### 1.3 PROJECT ACCOMPLISHMENTS

In this research, the distributed multitarget tracking approach developed in the last two DSN projects [1, 2, 3] was applied to acoustic tracking. The functional architecture for each node in the DSN remains the same. Each node contains the same modules of local processing, information fusion, information distribution, and possibly resource allocation.

An algorithm for tracking multiple air targets in a DSN of acoustic sensors was developed. The algorithm was based on a general multiple hypothesis approach to distributed multitarget tracking. Several modifications had to be made to accommodate the special characteristics of acoustic tracking discussed before. Since each sensor generates azimuths only, each node has two types of tracks, local tracks that are initiated locally and global tracks that are initiated from two cooperating nodes. Unresolved measurements from two or more targets due to poor sensor resolution was handled using a model for merging measurements. The model uses the sound pressure to assist in resolving targets.

Propagation delay implies that ordinary triangulation can not be used to initiate a global track (with location) from local tracks (with only azimuths) from two nodes. An algorithm that used both acoustic azimuths and azimuth rates was developed to estimate the target position and velocity. The communication among nodes is based on an adaptive strategy that only provides the recipient with useful information.

A simulation environment was developed on a Symbolics LISP machine for testing, evaluation, and demonstration of the algorithm. The environment includes a data generator that uses the same sensor models provided by Lincoln Laboratory. System architectures with different sensor numbers and geometries can be simulated. The communication pattern between the nodes can also be specified. In addition, pre-recorded real data can be read from a file and used to drive the processing algorithms. Graphical displays are provided to display the tracks for the different nodes as well as the intermediate results in the processing. Statistics can be collected during a run for performance evaluation.

The algorithm was implemented in the ADS simulation environment. About 20 different scenarios ranging from one to three targets and three to seven sensors were simulated. Both maneuvering and non-maneuvering targets were considered. The simulation results indicated that cooperation between nodes is essential but reasonable performance can be accomplished without continuous communication. Monte Carlo simulations were also performed for some scenarios. It was found that sensor resolution has significant impact on tracking performance, both in tracking accuracy and in data association performance. Limited real data were also obtained from Lincoln Laboratory to test the algorithms.

These results demonstrated that the distributed tracking approach developed by ADS in the DSN projects can be applied to a complicated sensor system consisting of acoustic sensors. Performance enhancements can probably be obtained if other types of sensors (e.g., radars, electro-optical) are available. The general nature of the basic algorithm allows the incorporation of these new sensors without much difficulty. On the other hand, the communication strategies can be improved to include requests for help, etc.

#### 1.4 REPORT ORGANIZATION

The rest of this report is organized as follows. Section 2 describes the general approach to distributed tracking in a DSN. This is mostly a review of the work done in earlier DSN projects. The basic structure will be used for acoustic sensors but the algorithms will have to be modified to accommodate the special features of acoustic tracking such as angle-only measurements, propagation delays, poor resolution, etc.

Section 3 describes the acoustic tracking scenario to be considered and the mathematical models used. The models reflect the special characteristics of acoustic tracking such as propagation delay and poor sensor resolution.

In Section 4, the local data processing algorithm adapted for acoustic tracking is presented in detail. An upgrade of the algorithm using multiple models, which significantly improves the local tracking performance, is also described.

Section 5 presents the information fusion and distribution modules of the system. Detailed algorithms for track-to-track fusion and likelihoods calculation are described. The strategies used for communication are also discussed.

Experimental results using simulated and real data are presented in Section 6. Evaluations of the performance by means of simulations and Monte-Carlo runs are also discussed.

Section 7 contains the conclusions and suggestions for future research. Appendices A and B present detailed derivations of some equations, while Appendix C is the report by Qualcomm, Inc. on tracking from a communication point of view.

## 2. DISTRIBUTED MULTITARGET TRACKING

This section contains a review of our approach to distributed multitarget tracking. Section 2.1 presents the architecture for each node in the DSN. Section 2.2 describes the local hypothesis processing functions. Section 2.3 describes hypothesis processing in information fusion. An approach to construct the information graph used in information fusion is presented in Section 2.4.

### 2.1 NODAL STRUCTURE

The ADS approach treats the DSN as a distributed hypothesis processing system. The overall goal of the system is to form hypotheses on the measurement data so that the targets can be tracked. Each node uses the local sensor data to form local hypotheses. When these are communicated to other nodes, new hypotheses are formed at the receiving nodes. Figure 2-1 shows the functional structure of each node and results from integrating Figures 1-2 to 1-4. Each node contains a local data base of hypotheses which is updated whenever new information arrives. This can happen in either one of two ways: data arriving from the local sensors or messages arriving from the other nodes. The two corresponding updating functions are then local information processing and information fusion.

#### 2.1.1 Local tracking data

Each node stores information about the targets in the form of hypotheses. These are formed from sensor reports received directly from local sensors or indirectly through other nodes. Each hypothesis is a possible explanation of the origins of the measurements in terms of how they are associated to the targets. Since multiple associations of the measurements to targets are possible, at any particular time a node maintains a set of hypotheses corresponding to the multiple explanations.

The relationship of a hypothesis to the measurements in the sensor reports is shown in Figure 2-2. Each hypothesis consists of tracks corresponding to the targets detected by the node. Each track consists of measurement indices from the same target. If the sensor resolution is such that a target cannot give rise to

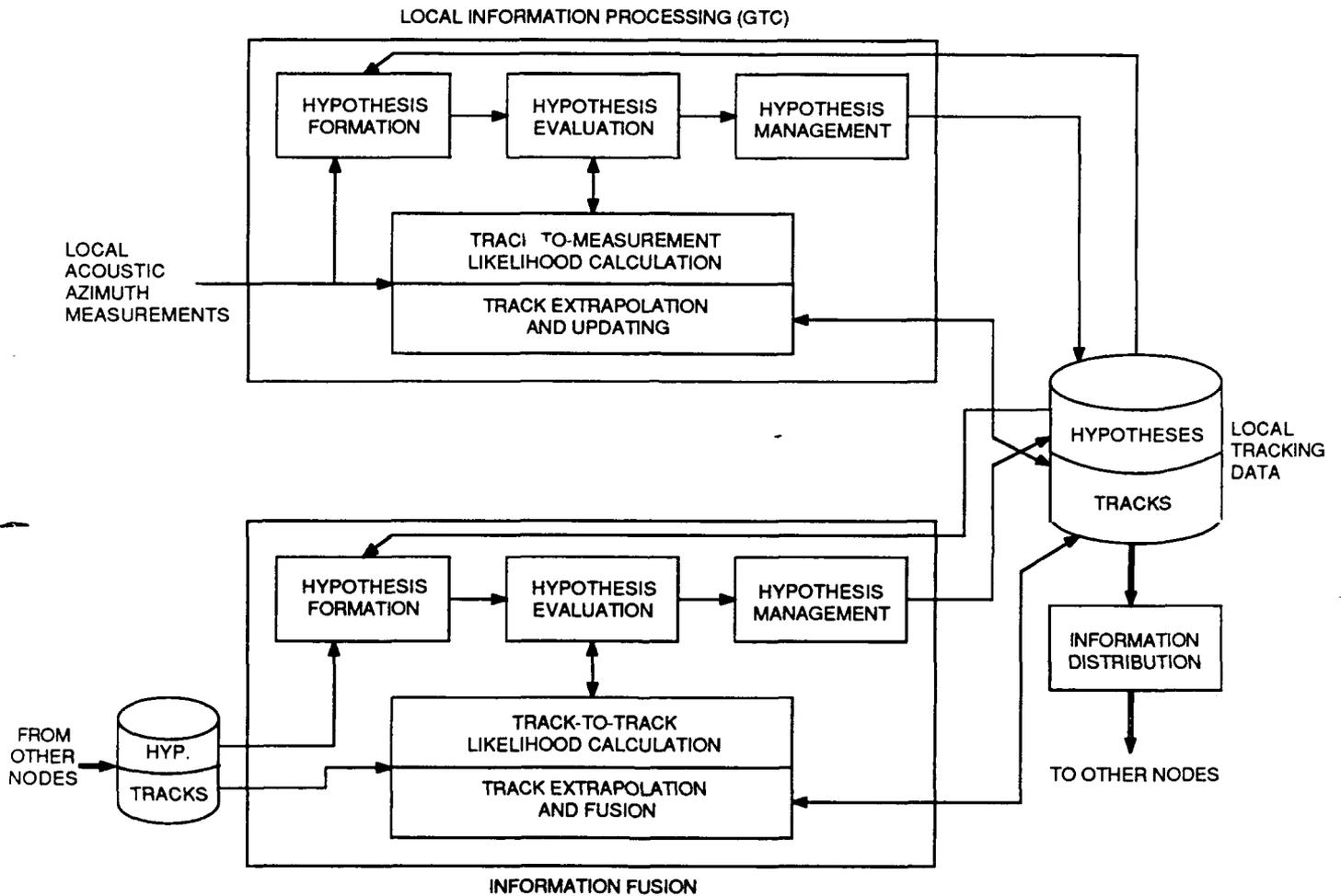


Figure 2-1: Functional Structure of Each Node

two or more measurements in the same sensor report, then this constraint would be used in defining a track. For acoustic sensors, this assumption that a target does not generate two or more measurements is generally true.

Many tracks can be formed from the measurement indices in the same sensor reports. However, not all of them may belong to the same hypothesis. A hypothesis is a collection of mutually consistent tracks. Consistency may depend on the sensor characteristics. If the sensor resolution is such that there are no merged measurements, then a possible hypothesis cannot have overlapping tracks. However, in the case of acoustic sensors, merged measurements are quite possible

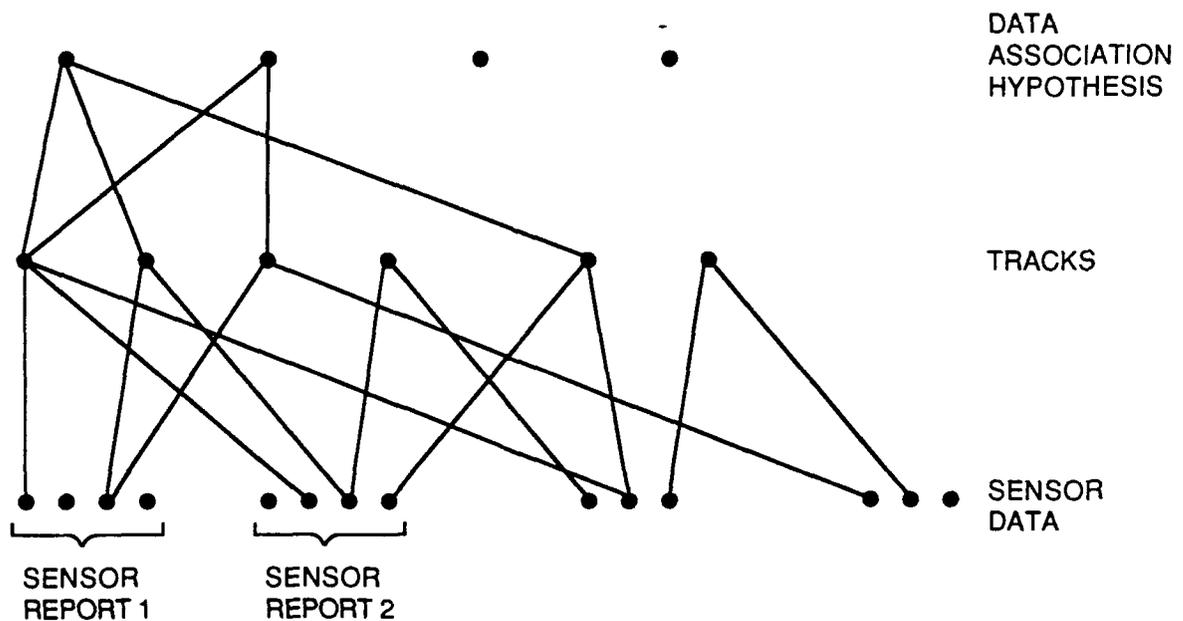


Figure 2-2: Definition of Hypothesis

when the target azimuths are close together. In any hypothesis, the measurements not belonging to any tracks are considered to be false alarms. A hypothesis also has associated with it a probability. Since each hypothesis set consists of mutually exclusive and collectively exhaustive explanations, the sum of the probabilities over the set would be one.

Once a track is specified, the measurements for the hypothesized target represented by the track are known. From these measurements, the state of the hypothesized targets can be estimated. For the targets of interest in the current research, the state consists of the position and velocity in an appropriate coordinate system. Since the measurements may contain error, the state uncertainty in terms of means and covariances should also be specified. In some cases, the classification of the target may also be included as a discrete state.

In a distributed framework, it is necessary to maintain the hypothesis sets formed or received at some earlier times in addition to the most current hypothesis set. The precedence relationship among the hypotheses and the tracks in the different sets is also needed. The hypothesis structure thus consists of multiple hypotheses at different times organized in the form of a directed graph. Figure 2-3 contains an example for periodic broadcast communication where the hypothesis set formed after the last broadcast time is also retained. The precedence relationship between the hypotheses is shown by means of links between the hypotheses. Although not shown, there should be similar links between the tracks. In general, the dimension of the hypothesis structure depends on the communication between the nodes.

### **2.1.2 Processing structure**

The hypothesis structure presented above represents the information state for each node. This information state is updated whenever new information is received in the form of sensor data or messages from other nodes. Functionally, the node contains subsystems or modules responsible for local information processing, information fusion and information distribution. The local information processing module updates the hypothesis structure with the local sensor data. The information fusion module updates the hypothesis structure with incoming messages from other nodes. The information distribution module is responsible for communication with other nodes.

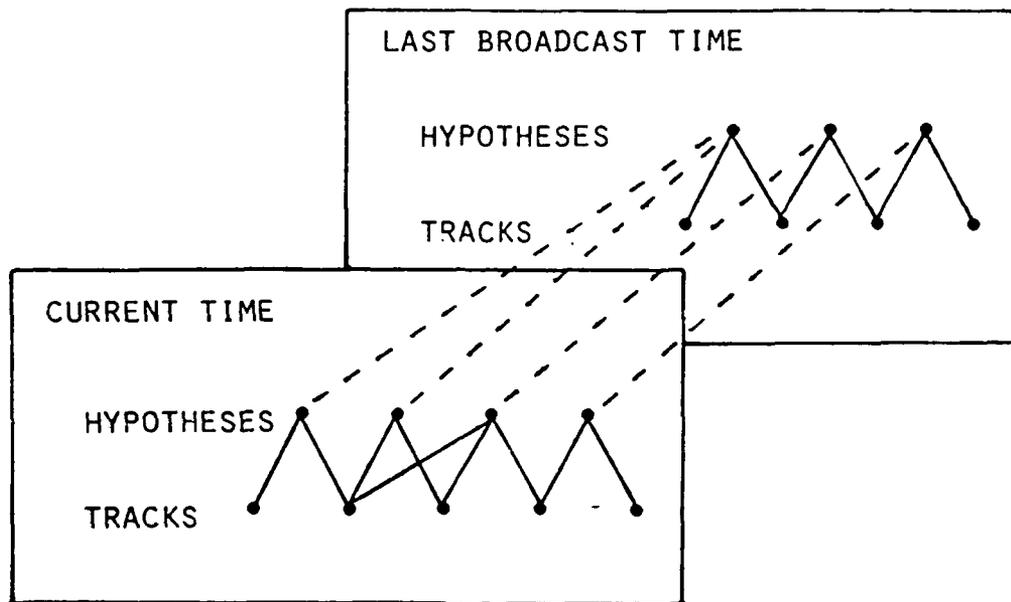


Figure 2-3: Example of Hypothesis Structure

The local information processing and information fusion modules contain two levels as in Figure 2-1. The first level is concerned with hypothesis processing functions such as their formation, evaluation and management. Of these, hypothesis evaluation requires information at the track level. There are thus additional track-level calculations to be performed in each module. For local information processing, they involve calculating the track-to-measurement association likelihoods and track state distributions. For information fusion, the corresponding calculations would involve the track-to-track association likelihoods and track state distributions. In general the hypothesis processing functions are almost the same for many applications. They will be considered in the following

subsections for both local information processing and information fusion. The track-level calculations, on the other hand, are dependent on the particular target and sensor scenarios. They will be discussed in details, specifically for acoustic tracking, in Sections 4 and 5.

## **2.2 LOCAL HYPOTHESIS PROCESSING**

The local information processing module uses data (acoustic azimuth measurements) from the local sensors to generate new hypotheses from the old hypotheses. It performs the following functions:

- Hypothesis formation
- Hypothesis evaluation
- Hypothesis management

Of these functions, hypothesis formation and management are at the hypothesis level while hypothesis evaluation also requires inputs from track-level calculations. The hypothesis processing functions will be described in this section while the track-level calculations will be described in Section 4.

### **2.2.1 Hypothesis formation**

This is the first step in local hypothesis processing. New hypotheses are formed by associating the measurements in each sensor report with the tracks in the current hypothesis set. Multiple associations are in general possible for each hypothesis. The following are the different ways in which measurements and tracks can be associated:

- A measurement  $y_j$  may be associated with an existing track  $\tau_i$  in the current hypothesis
- A measurement  $y_j$  may be associated with a new track (i.e., a target which has not been detected up to now)
- A measurement  $y_j$  may be associated with no track at all (i.e., it may come from a false alarm)
- A track  $\tau_i$  may not be associated with any measurement at all (i.e., it may be missed in the current sensor report)

Not all of these associations may be meaningful and some initial screening is used to reduce the possible associations. For example, if a measurement and a track are too far apart in location, then there should not be any association. From these possible associations, new tracks are formed by appending the measurements to the old tracks. From the new tracks, new hypotheses are formed by imposing suitable constraints among the tracks in the same hypothesis. If the sensor resolution is such that there are no merged measurements, then there should be no overlap in the tracks. New hypotheses are generated from tracks satisfying these constraints.

There are several ways to perform the actual hypothesis formation. The simplest is by a recursive list method. Consider the example in Table 2-1. It displays the possible associations between measurements (on the x-axis) from one sensor report and tracks (on the y-axis) in a hypothesis. The sensor report consists of two measurements: 1 and 2. 0 represents the absence of a measurement, or non-detection. There are two existing tracks: 1 and 2.  $N$  represents a new track and  $F$  represents false alarms. Thus either one of the measurements can be new tracks or false alarms. *Track 2*, however, cannot be associated with *measurement 1* since the value of *measurement 1* is incompatible with the state of *track 2*. The following steps can be used for hypothesis formation:

Table 2-1: Track-Measurement Cross Reference Table

	0	1	2
N	X	X	X
1	X	X	X
2	X	O	X
F		X	X

1. For each measurement in the table, the column below it gives the list of possible tracks which can be associated with it. The lists for the two measurements are:

- measurement 1:  $(N,1,F)$

- measurement 2:  $(N,1,2,F)$

2. By forming all possible combinations between list 1 and 2, with one track from each list, we obtain twelve possibilities of the form  $(a,b)$  where  $a$  is the track associated with *measurement 1* and  $b$  is the track associated with *measurement 2*. Of these,  $(1,1)$  is eliminated since *track 1* cannot be both associated with *measurement 1* and *2* in the same sensor report. There are thus eleven possible hypotheses.

The eleven hypotheses are displayed in Table 2-2. Figure 2-4 shows the hypothesis set in the form of a tree. Each branch of a tree represents a possible hypothesis. The symbol under each measurement denotes the track associated with the measurement in that particular hypothesis. In this approach, each measurement is associated to at most one track. It thus guarantees that there are no merged measurements. However, some combinations will involve split measurements and have to be eliminated.

Table 2-2: Hypothesis Set (no merged measurements)

MISSED TRACK	$\phi$ 2 2 2 1 {1, 2}{1, 2} 2 1 {1, 2}{1, 2}
TRACK FOR MEASUREMENT 1	1 1 1 F F F N N N N
TRACK FOR MEASUREMENT 2	2 F N 1 2 F N 1 2 F N

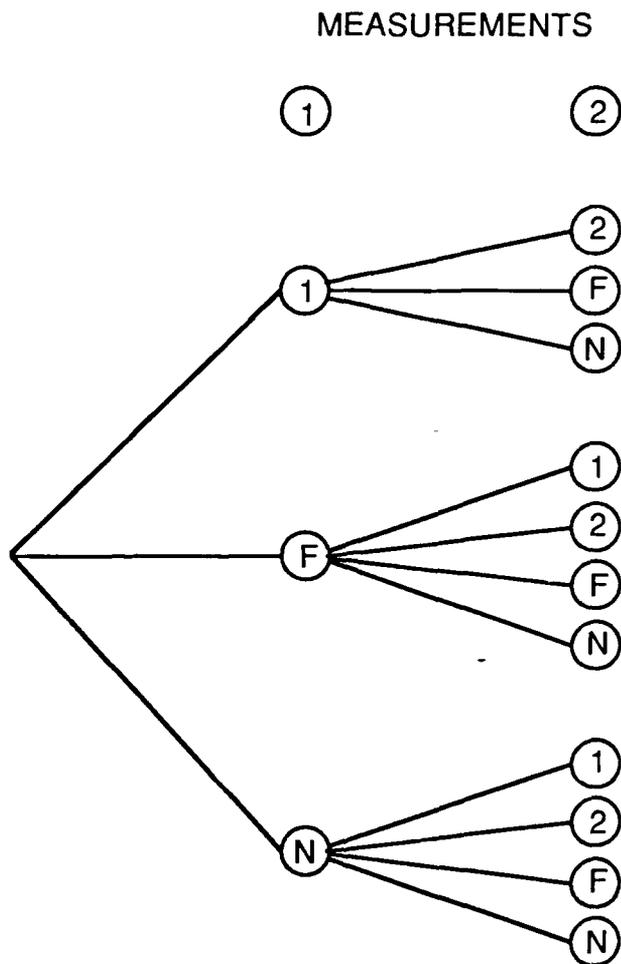


Figure 2-4: Hypothesis Tree (no merged measurements)

For the scenario in the current research, however, the acoustic sensor resolution is such that two or more detected targets may generate only one measurement. In this case the hypothesis expansion scheme would have to be modified. Specifically, each measurement can be associated with two tracks. However, the no track splitting assumption should still be valid, i.e., the same track should not contain more than one measurement in the same sensor report. The recursive list technique used above needs to be modified to allow merged measurements. But first of all, the track-to-measurement correlation (TMCR) table needs to be

modified according to the merged measurement model.

For the same example as in Table 2-1, suppose *measurement 2* could be a merged measurement from the two targets. Then the modified (Extended) TMCR table can be obtained as in Table 2-3. In this table, an extra row hypothesizing the merging of measurements from *tracks 1 and 2* is added. This row contains a non-zero entry at column 2 which represents the possibility of *measurement 2* being the merged measurement according to the model. In this case, the following steps can be used for hypothesis formation :

1. For the two measurements in the table, the new lists for the possible tracks are:

- measurement 1:  $(N,1,F)$

- measurement 2:  $(N,1,2,\{1,2\},F)$  -

Table 2-3: Extended Track-Measurement Cross Reference Table

	0	1	2
N	X	X	X
1	X	X	X
2	X	O	X
{1,2}		O	X
F		X	X

2. By forming all possible combinations between list 1 and 2, with one track from each list, we obtain fifteen combinations. Of these, (1,1) is eliminated as before, similarly (1, {1,2}) is also eliminated based on the same reason. There are thus thirteen possible hypotheses left.

Table 2-4 contains the hypothesis set. Figure 2-5 displays the same hypothesis set as a tree. Note that in this case more than one track can be associated to the same measurement.

Note that this approach of hypothesis formation automatically guarantees that there are no split measurements for existing tracks. Hypotheses with merged measurements are formed if no additional constraints are imposed. On the other hand, one can remove the hypotheses with overlapping tracks to obtain the hypothesis set in Table 2-2.

### 2.2.2 Hypothesis Evaluation

This module evaluates the probability of each hypothesis being true. A recursive algorithm has been developed in previous projects [1, 2, 3] for hypothesis evaluation. Given a hypothesis  $\lambda$  which descends from another hypothesis  $\bar{\lambda}$  in the sense that all the tracks in the current hypothesis are either extensions of the old tracks or new tracks, the hypothesis evaluation algorithm is given by:

Table 2-4: Hypothesis Set (with merged measurements)

MISSED TRACK	$\phi$ 2 2 2 1 $\phi$ {1,2}{1,2} 2 1 $\phi$ {1,2}{1,2}
TRACK FOR MEASUREMENT 1	1 1 1 F F    F    F    F    N N N    N    N
TRACK FOR MEASUREMENT 2	2 F N 1 2 {1,2}    F    N    1 2 {1,2}    F    N

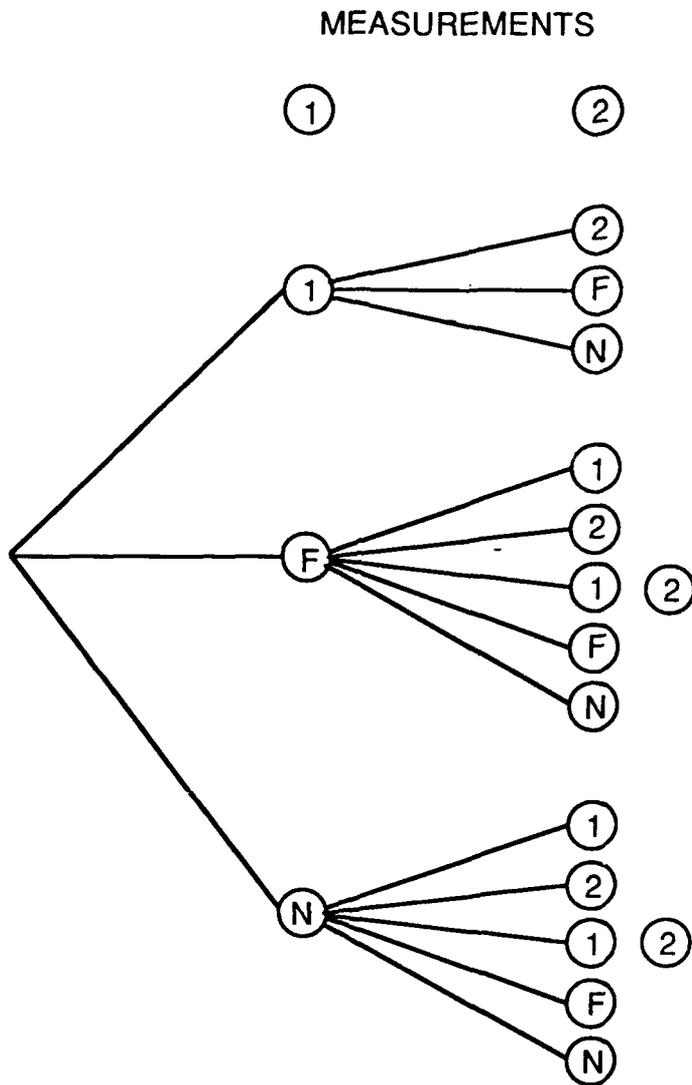


Figure 2-5: Hypothesis Tree (with merged measurements)

$$P(\lambda | Z) = C^{-1} P(\bar{\lambda} | \bar{Z}) L_{FA} \prod_{\tau \in \lambda} L(\bar{\tau}, y_{\alpha(\tau)}) \quad (2.1)$$

where  $Z$  is the cumulative data up to the current time,  $\bar{Z}$  is the cumulative data up to the previous update time, and  $C$  is a normalizing constant.  $L_{FA}$  is the likelihood of false alarms in the hypothesis.  $L(\bar{\tau}, y_{\alpha(\tau)})$  is the likelihood that a track  $\bar{\tau}$  is associated with the measurement  $y_{\alpha(\tau)}$ . The exact expressions of these likelihood functions can be found in previous DSN reports.

Note that in addition to the probability of the previous hypothesis, certain track-to-measurement likelihoods are needed to compute the probability of the current hypothesis. Specifically, the following likelihoods of associations are needed:

- Likelihood of a track associated with a measurement
- Likelihood of a track associated with no measurement
- Likelihood of a measurement associated with a new track
- Likelihood of a measurement being a false alarm

In addition to the measurement values and track state values, each of these likelihoods depends on the measurement errors and state estimate errors. They also depend on the detection probability of the sensor. Note that when all the hypotheses have to be evaluated, the likelihoods needed can be summarized conveniently in a track-to-measurement cross reference table (Table 2-5). On one dimension of the table are all the tracks 1 to  $N_t$ , the new track denoted by 0 and the false alarm. On the other dimension are the measurements 1 to  $N_m$  in the current sensor report. The measurement indexed by 0 corresponds to nondetection. Each element in the table is a likelihood associating a track (old, new, or false alarm) to a measurement (or nondetection). These likelihoods have to be provided by track-level calculations to be discussed in later sections.

Table 2-5: Track-Measurement Cross Reference Table

		MEASUREMENTS				
		0	1	2	• • •	$N_m$
New Track	0	$L_{00}$	$L_{01}$	$L_{02}$	• • •	$L_1 N_m$
	1	$L_{10}$	$L_{11}$	$L_{12}$	• • •	$L_1 N_m$
Existing Tracks	•	•	•	•	$L_{ij}$	•
	•	•	•	•		•
	•	•	•	•		•
	$N_t$	$L_{N_t,0}$	$L_{N_t,1}$	$L_{N_t,2}$	• • •	$L_{N_t, N_m}$
False Alarm			$L_{F1}$	$L_{F2}$	• • •	$L_{FN_m}$

The hypothesis evaluation algorithm presented above assumes that there are no merged measurements. When this is not the case, it will be necessary to consider the likelihood of two or more targets generating the same measurement. The detailed hypothesis evaluation for merged measurements will be discussed in Section 4.

### 2.2.3 Hypothesis management

The function of the hypothesis management module is to control the number of hypotheses maintained in the system. The following hypothesis management techniques have been implemented in the current ADS DSN algorithms:

- Hypothesis pruning: Hypotheses whose probabilities are below a certain threshold are removed from further consideration. Another alternative is to retain enough hypotheses so that their cumulative probability is above a threshold. This is called the adaptive thresholding approach.

- Hypothesis combining: Similar hypotheses are combined into a single hypothesis. Similar hypotheses are those which have the same number of tracks and whose tracks are similar according to some criterion.
- Hypothesis clustering: Groups of measurements and tracks which cannot be associated can be decomposed into independent clusters. Hypothesis formation and evaluation can then be performed within each cluster. This reduces the amount of storage and computation without making any approximation.

These techniques are used for managing the number of hypotheses in the current system.

## 2.3 HYPOTHESIS PROCESSING IN FUSION

The information fusion module takes hypotheses received from other nodes and integrates these with the local hypotheses to generate new hypotheses. A hypothesis from each node consists of target tracks which the node is supposed to have detected. Fusion produces new hypotheses consisting of detected target tracks given the local and received information. It is basically a track correlation process, i.e., determining which tracks from the different nodes correspond to the same targets. As in local hypothesis processing, there are three separate steps in information fusion: hypothesis formation, hypothesis evaluation and hypothesis management. Before considering these topics, the issue of track and hypothesis fusability will be discussed.

### 2.3.1 Fusability and information graph

The first step in the fusion process is to form possible track and hypothesis sets using the local tracks and hypotheses and those received from other nodes. Certain combination of tracks and hypotheses should not be fused since they are inherently contradictory. In the example of Figure 2-6, the two tracks  $\tau_1$  and  $\tau_2$  are two local tracks maintained at two different nodes. They cannot be fused since the resulting global track would have two different measurements in the same sensor report 1, thus violating the no split measurement assumption. On

the other hand,  $\tau_1$  and  $\tau_3$  can be fused to yield a global track  $\tau_1 \cup \tau_3$ . The interpretation of this global track is that the measurements in both tracks  $\tau_1$  and  $\tau_3$  come from the same target. Tracks  $\tau_1$  and  $\tau_4$  can also be fused. However, they do not have to be and in that case the two tracks correspond to two different targets. The fusability question also needs to be addressed at the hypothesis level. Each local hypothesis is a possible explanation about the origins of the local measurements. Thus if the local hypotheses are incompatible, they cannot be fused to obtain a global hypothesis. This is illustrated in Figure 2-7 where each node  $i$  has two local hypotheses  $\lambda_i^j, j=1,2$  derived from the two common hypotheses  $\lambda^j, j=1,2$ . Since  $\lambda^1$  and  $\lambda^2$  are mutually exclusive, the local hypotheses  $\lambda_1^2$  and  $\lambda_2^1$  are not fusable.

In the examples above, it is easy to determine the fusability of hypotheses and tracks. When communication becomes more complicated, determination of fusability becomes more difficult since it is necessary to identify the information available to the nodes in the network at various times and how the information of one node at one time is related to that of another node at a different time. For example, whenever two nodes communicate some common information is shared between the nodes. The existence of this shared information would affect

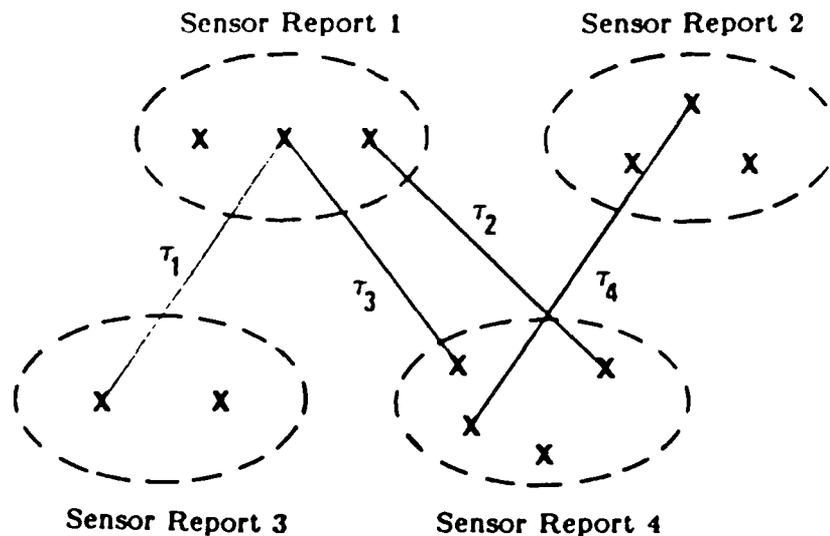


Figure 2-6: Fusability of Tracks

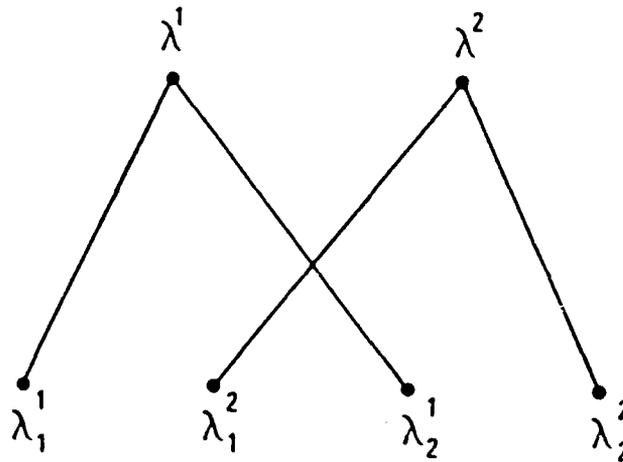
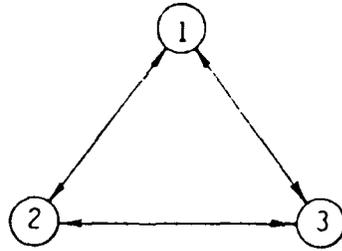


Figure 2-7: Fusability of Hypotheses

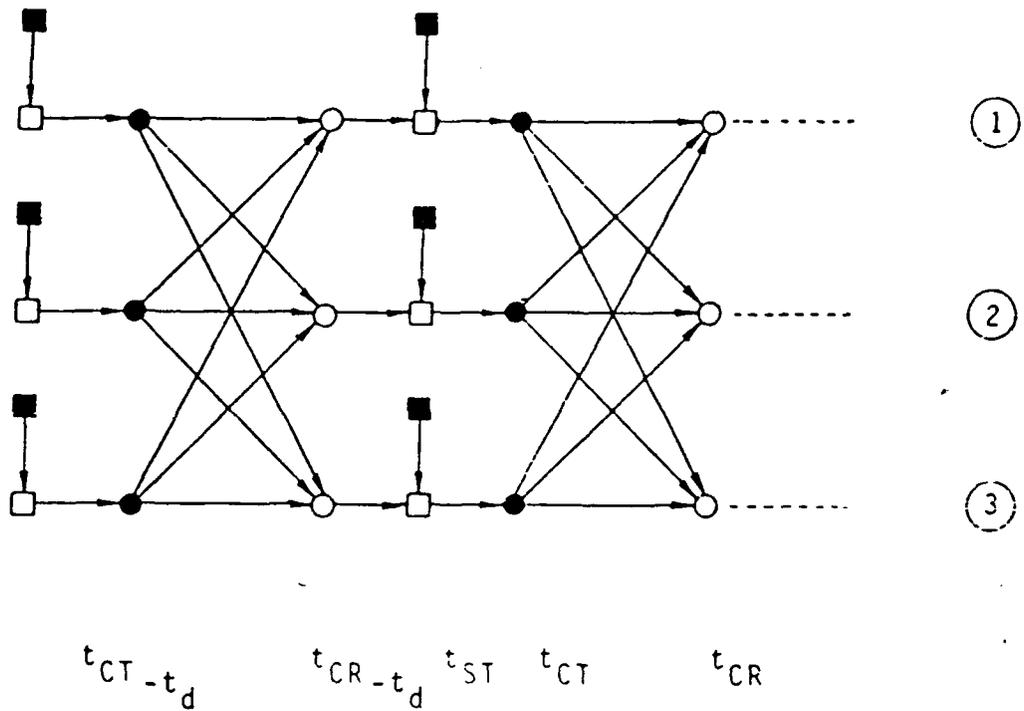
fusability since hypotheses which have different predecessors in this common information are not fusable.

The information graph can be used to trace the histories of the communication conveniently. Each vertex in the graph represents an event when the information in the DSN changes. These events are: sensor observation, sensor data received at a node, transmission of messages by a node and reception of messages at a node. The arc in the graph represents information flow. Figure 2-8 shows the information graph for broadcast communication. At a given time all the nodes communicate to each other so that they all have the same information. Figure 2-9 shows the information graph for a cyclic communication system. The system consists of three nodes  $N=\{1,2,3\}$  collecting data from the three sensors  $S=\{1,2,3\}$ , respectively at the times  $\dots, t_{ST}, t_{ST}+t_d, \dots$ . The nodes transmit to the other nodes periodically according to the pattern shown in Figure 2-9 at times  $\dots, t_{CT}, t_{CT}+t_d, \dots$  and the messages are received at the times  $\dots, t_{CR}, t_{CR}+t_d, \dots$ . It is assumed that  $t_{ST} < t_{CT} < t_{CR}$ .

SYSTEM



INFORMATION GRAPH



- Sensor Observation
- Communication Transmission
- Sensor Data Reception
- Communication Reception

Figure 2-8: Broadcast Communication

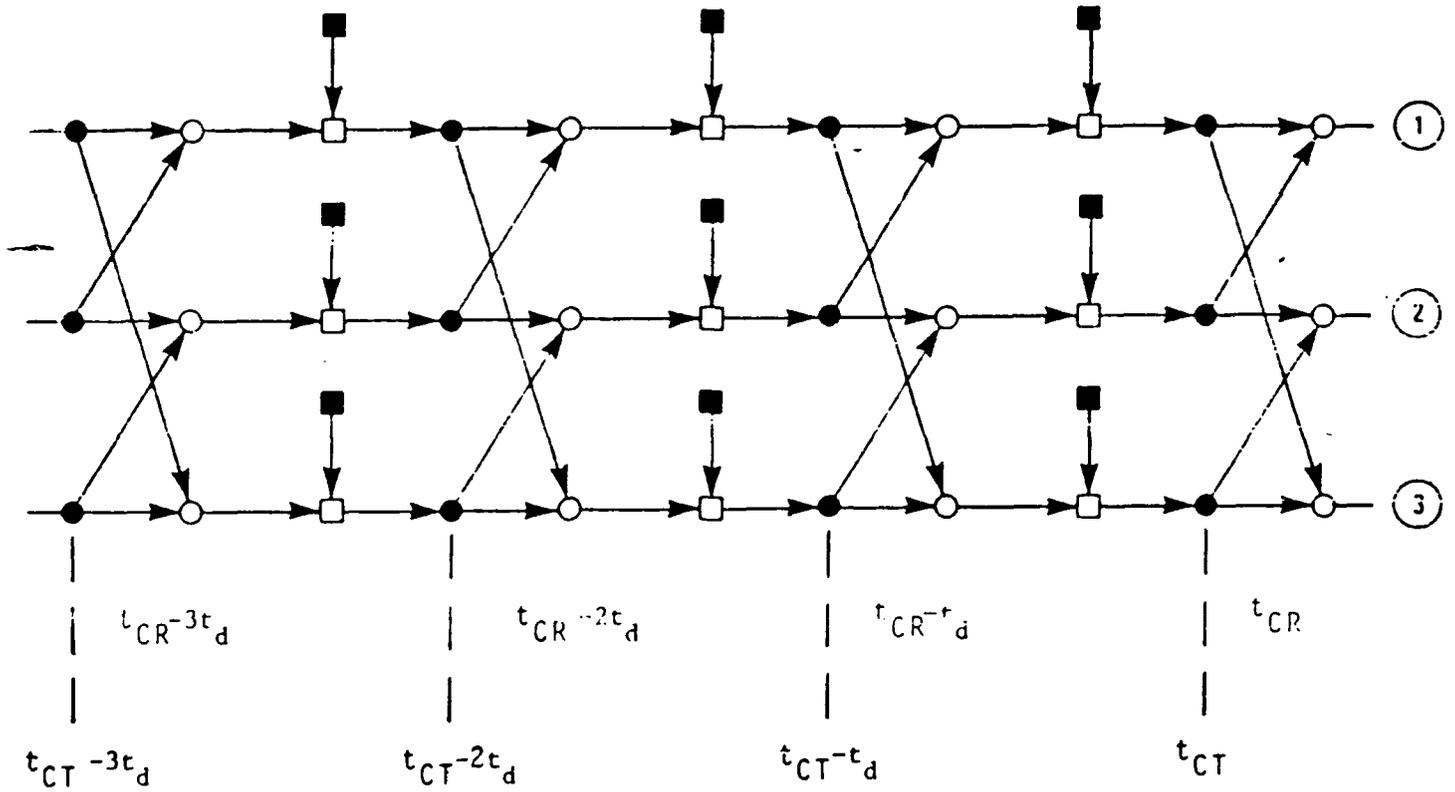
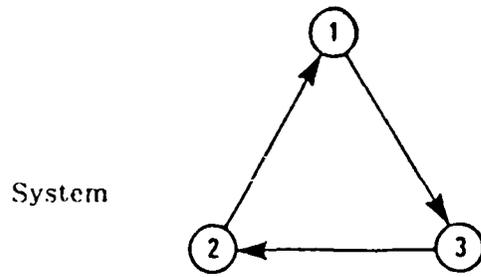


Figure 2-9: Cyclic Communication

### 2.3.2 Hypothesis formation

Once the information graph is available, The following steps are used in hypothesis formation:

1. Determining hypothesis fusability. Hypotheses which are inconsistent, e.g., descending from conflicting hypotheses, cannot be fused. Fusion amounts to checking the ancestors of the current hypotheses. If they are processed by a node at any given time, then they should have nonconflicting ancestors at that particular vertex in the information graph. In fact it is only necessary to check the most recent common predecessor set. In the example of Figure 2-9, consider the hypothesis sets at vertices  $(t_{CT},1)$  and  $(t_{CT},2)$  which are fused at the vertex  $(t_{CR},1)$ . A pair of hypotheses in these sets would be inconsistent if their ancestors on  $(t_{CT}-2t_d,1)$  or  $(t_{CT}-t_d,2)$  (or predecessors of these vertices) are different. A necessary condition is to check their ancestors on  $(t_{CT}-2t_d,1)$  or  $(t_{CT}-t_d,2)$ . If they are different, then the hypotheses are not fusable. Otherwise, we can proceed to the next step.
2. Determining track fusability. For each fusable hypothesis pair, all fusable track pairs are identified. A fusable track pair is one which when traced back does not cause any conflict. Non-fusable tracks are those which have been hypothesized to come from distinct targets in the past. In particular, all new tracks created after the nodes last communicated are fusable. The fusability condition again can be checked as in hypothesis fusability. However, the condition is now both necessary and sufficient [7]. A track-to-track fusability table such as Table 2-6 can be used to represent the fusable tracks. Each nonzero entry corresponds to a pair of fusable tracks. In Table 2-6,  $\tau_{21}$  and  $\tau_{11}$  are a fusable pair and neither track can be fused with any other track. This implies that they have been identified to have come from the same target in the past. The tracks  $\tau_{12}$  and  $\tau_{22}$ , however, are fusable with each other or with the undetected tracks represented by  $\tau_{i0}$ . This means that they can be the same targets or they can be different targets which are not detected by the other node. In forming the track-to-track fusability table, information about the track states can also be used. For example, if the tracks

Table 2-6: Track-to-Track Fusability Table

		$\lambda_2$		
		$\tau_{20}$	$\tau_{21}$	$\tau_{22}$
$\lambda_1$	$\tau_{10}$	X	O	X
	$\tau_{11}$	O	X	O
	$\tau_{12}$	X	O	X

$\tau_{22}$  and  $\tau_{12}$  are very far apart, then the two tracks should not be fused even though they are logically fusable.

3. Hypothesis formation for each fusable hypothesis pair. From the two hypotheses  $\lambda_1$  and  $\lambda_2$ , we can thus form the following two hypotheses:

$$\{(\tau_{11} \cup \tau_{21}), (\tau_{10} \cup \tau_{22}), (\tau_{12} \cup \tau_{20})\}$$

$$\{(\tau_{11} \cup \tau_{21}), (\tau_{12} \cup \tau_{22})\}$$

This step can be repeated for other fusable hypothesis pairs.

As a result of these three steps, we have multiple hypotheses, each one corresponding to a different association of the tracks between the nodes. Hypothesis formation is basically a two-level procedure: the top level considers all possible associations among the hypotheses, and given each of these the second level considers all fusable associations among the tracks. Note that the second level is actually very similar to hypothesis formation at the local level.

### 2.3.3 Hypothesis evaluation

The next step in information fusion is to evaluate the probabilities of the hypotheses formed using the probabilities of the local hypotheses and the local track state distributions. If the nodes communicated in the past, the local statistics would not be independent. A key problem in hypothesis evaluation is to identify the common information shared by the nodes and make sure it is not double counted in generating the global statistics.

The information graph is useful in tracing the shared information in the hypothesis probabilities and track state distributions. Suppose it is necessary to compute the conditional probability of the state  $x$  at a vertex  $i_0$  in the information graph whose immediate predecessors in the graph form the set  $I$ . Suppose the cumulative measurement set for an immediate predecessor  $i$  is  $Z_i$  and  $\bar{I}$  is the set of predecessors of  $I$ . Since each  $p(x | Z_i)$  contains some information shared by other  $p(x | Z_j)$ 's, this redundant information needs to be identified and removed in the fusion. It was shown in [7] that the probability of  $x$  at the fusion vertex is given by

$$p(x | \bigcup_{i \in I} Z_i) = C^{-1} \prod_{\bar{i} \in \bar{I}} p(x | Z_{\bar{i}})^{\alpha(\bar{i})} \quad (2.2)$$

where  $\bar{I} \subseteq I$  is a subset of  $\bar{I}$ ,  $(\alpha(\bar{i}))_{\bar{i} \in \bar{I}}$  is some index tuple such that  $\alpha(\bar{i})$  is a nonzero integer for each  $\bar{i}$ , and  $C$  is the normalizing constant. The set  $\bar{I}$  contains all the information vertices which are relevant to fusion at the vertex  $i_0$ .  $\alpha(\bar{i})$  determines whether the information at vertex  $\bar{i}$  should be added ( $\alpha(\bar{i})=1$ ) or removed ( $\alpha(\bar{i})=-1$ ). In equation (2.2), addition of information appears as multiplication by the conditional probability while removal appears as division.

Let  $Z = \bigcup_{i \in I} Z_i$  be the cumulative measurement data after fusion has taken place. We need to compute the probability  $P(\lambda | Z)$  for each hypothesis  $\lambda$  formed from fusion. For the tracking scenario under consideration, the following hypothesis evaluation algorithm is applicable. Suppose for each  $\bar{i} \in \bar{I}$ , the probability  $P(\lambda | Z_{\bar{i}})$  for each hypothesis  $\lambda$  defined on  $Z_{\bar{i}}$  is known. Then for each hypothesis  $\lambda$  defined on  $Z$ , the probability of the hypothesis being true is given by

$$P(\lambda | Z) = C^{-1} \prod_{\bar{i} \in \bar{I}} P((\lambda | J_{\bar{i}}) | Z_{\bar{i}})^{\alpha(\bar{i})} \prod_{\tau \in (\lambda | J)} \tilde{L}(\tau, (Z_{\bar{i}})_{\bar{i} \in \bar{I}}) \quad (2.3)$$

where  $C$  is a normalization constant,  $(\lambda | J_{\bar{i}})$  is the predecessor hypothesis of  $\lambda$  defined on  $Z_{\bar{i}}$  and  $\tilde{L}(\tau, (Z_{\bar{i}})_{\bar{i} \in \bar{I}})$  is the likelihood of the global track formed by associating the local tracks. The exact formula of this track-to-track association likelihood was derived in [7] and will be discussed again in Section 5.

We note that hypothesis evaluation depends only on the statistics at the information vertices in the set  $\bar{I}$ . The function  $\alpha$  determines whether the information at a vertex should be added or subtracted. The hypothesis evaluation formula of (2.3) has a two-level structure. At the higher level, the product of the local hypothesis probabilities evaluates the probability of associating the given set of local hypotheses. The next level consists of the likelihoods of associating the individual tracks. As in local hypothesis evaluation, these track-to-track association likelihoods can again be supplied in the form of a table. Section 5 will discuss the calculation of these likelihoods.

#### 2.3.4 Hypothesis management

Hypothesis management techniques such as those used in the local information processing would be needed in information fusion. Examples are pruning, combining and clustering. Pruning and combining are usually performed within each cluster.

### 2.4 CONSTRUCTION OF INFORMATION GRAPH

In Section 2.3, we discussed the information fusion problem. Both hypothesis formation and hypothesis evaluation requires knowing the information graph so that fusable hypotheses and tracks can be identified and redundant use of information in either hypothesis evaluation or track fusion can be avoided. When the communication schedule is specified a priori, the information graph can be generated off-line, stored at each node and used in information fusion. When communication is driven by data or unreliable, the information graph cannot be generated in advance. Techniques used to construct the information graph on-line will be discussed in this section.

### 2.4.1 Limited range broadcast communication

The nominal communication in the acoustic tracking scenario is limited range broadcast where each node broadcasts to only a finite number of neighbors within the broadcast range. Consider the example in Figure 2-10. The system has six nodes arranged as in the figure and a node can only hear from one which is connected to it. The information graph depends on the order of their broadcast and cannot be specified a priori. Suppose they broadcast sequentially in the order of 1,2,3,4,5,6. The information flow graph is shown in Figure 2-11. When they broadcast in a different order, such as 1,3,5,2,4,6, the information graph will be different.

A possible way of generating the information graph dynamically is to attach a history to each hypothesis set communicated in the system. When fusion occurs, each DSN node appends the histories of the fused hypotheses to the resulting hypothesis set. This results in the transmission of the history of the hypothesis set together with the hypothesis set. As an example, after fusion by node 2 at time  $t_2$ , the resulting hypothesis set has a tag indicating its ancestor from node 1. After fusion by node 2 at time  $t_4$ , the hypothesis set would know that it only has data from nodes 1 and 2. The hypothesis sets maintained by nodes 1, 5 and 6 just before  $t_{12}$  would have the history represented in Figure 2-12. When node 1 receives the broadcast from node 6, the histories of the local and incoming hypothesis sets contain the partial information graphs at the two nodes. This can be used to construct the relevant part of the information graph needed for information fusion. The common information shared by the two nodes is that of the vertex  $(t_3, 2)$ . This is the last time the nodes communicated with one

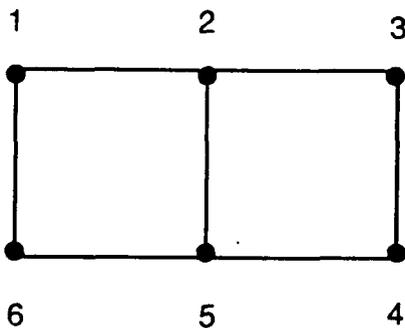


Figure 2-10: Six Node Configuration

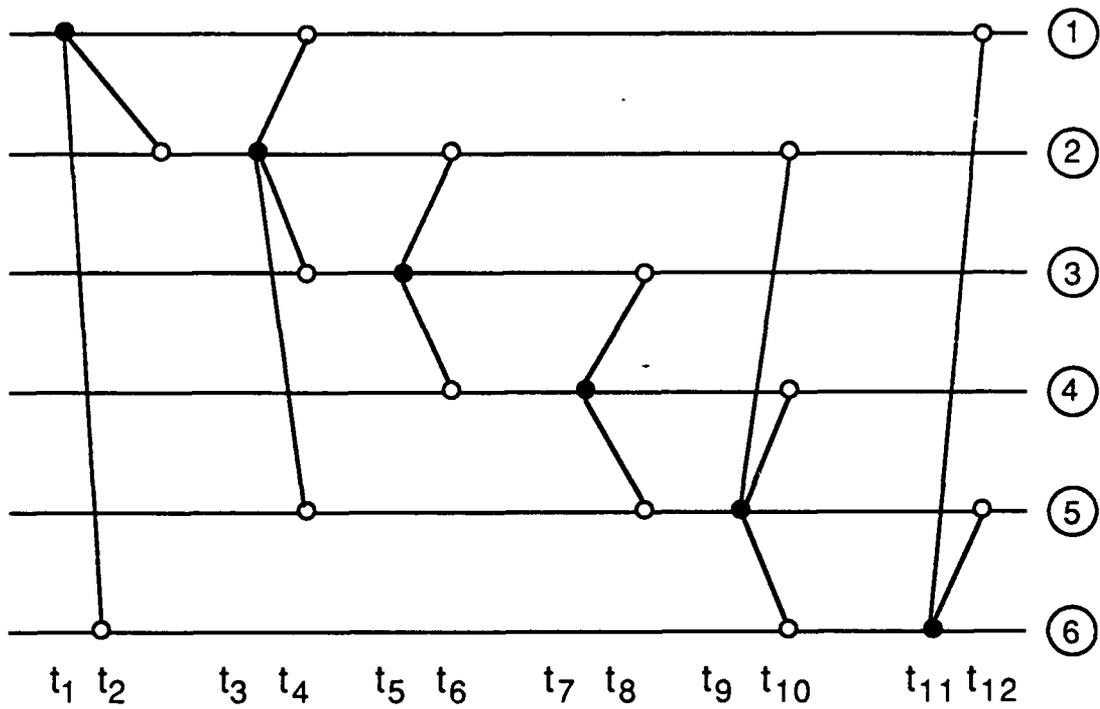


Figure 2-11: Information Graph for Sequential Broadcast

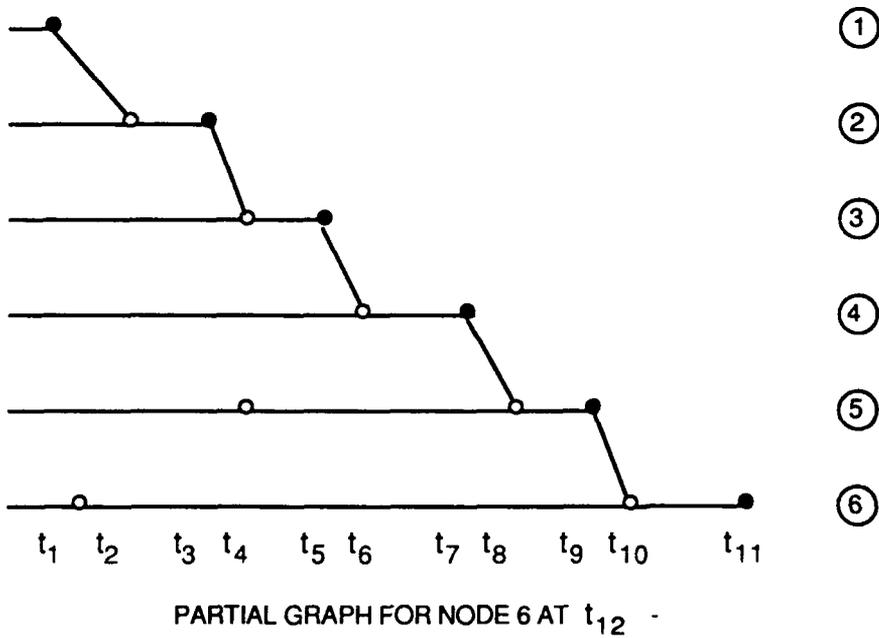
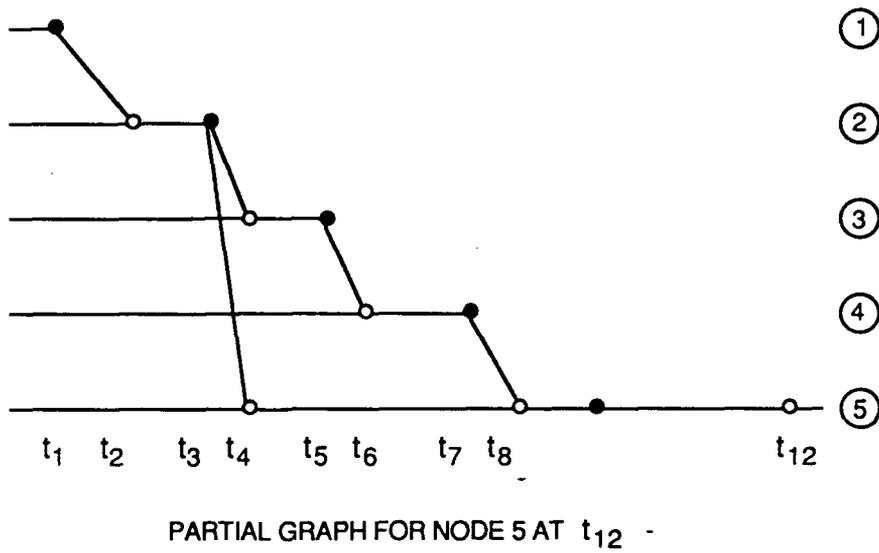
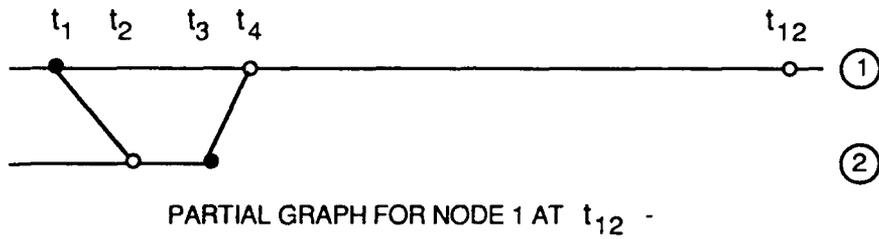


Figure 2-12: Partial Information Graphs

another. Hypothesis and track fusability require checking the ancestors of hypotheses and tracks at this vertex. Both hypothesis evaluation and track state fusion would use the probabilities at this vertex. Similarly, node 5 can construct the relevant part of the information graph by using the partial information graphs from node 5 and node 6. In this case, the shared information is at the vertex  $(t_9,5)$ .

Some approximations are needed to make the scheme practical. The history for each hypothesis set cannot be too long or the communication and memory resources will be too high. In addition, the reconstruction of the information graph may be too difficult. Some kind of window needs to be applied to the history of each hypothesis set to reduce it to a manageable length. In some cases the finite window will not introduce any error. For example, if all the information before  $t_8$  in Figure 2-12 is discarded, node 5 would still be able to identify the common information vertex  $(t_9,5)$ . On the other hand, node 1 would not be able to trace the shared information in hypothesis sets from node 1 and node 6. However, one may argue that any dependence which occurs in the distant past should not have too much effect on the current processing. These communication issues will be discussed further in Section 5.

The complexity of the problem is sometimes reduced if each node does not process all the messages received. For example, if the messages contain information about tracks which are not yet in the sensor's field of view, then they need not be included in the processing. This simplifies the information graph since a node which does not use the message is equivalent to one which does not receive the message.

#### **2.4.2 Lost messages and communication failures**

Since communication is not always reliable, broadcast messages may be lost. The broadcasting node may not be aware of this if no acknowledgement is provided in the system. In addition, some nodes in the broadcast range may receive the message while others may not. Since the effective information graph is modified, information fusion will be affected. For example, consider figure 2-11. Suppose node 6 does not receive the broadcast message from node 5 at time  $t_9$ . The path from  $(t_9,5)$  to  $(t_{10},6)$  in the information graph is then absent. When fusion at node 5 occurs at time  $(t_{12},5)$ , the common information is now  $(t_1,1)$ . If

this is not recognized, error may be introduced into the processing. This problem, however, would not arise if the information graph is constructed on-line from the histories of the communication.

### 3. DISTRIBUTED ACOUSTIC SYSTEM AND MODELS

In this section, we describe the distributed acoustic tracking system and the models used. Section 3.1 presents the overview of the system, and Section 3.2 describes the models.

#### 3.1 SYSTEM DESCRIPTION

Our system configuration is based upon the one implemented by Lincoln Laboratory in their DSN testbed.

The acoustic sensors in the DSN test bed are small microphone arrays. The front-end signal processing algorithms produce "measurements" every two seconds and they correspond to the average target azimuths over the two second interval. The algorithms also supply signal-to-noise estimates, which can be used to generate accuracy values for the azimuth measurements. The sensors produce no target elevation information.

The detection range for a single target is from a few to a few tens of kilometers, with five kilometers being a good nominal value. The target detection probability depends upon the signal-to-noise ratio, which for a given signal source strength depends upon range, topography, background noise, and propagation conditions. In general, detection probability increases with decreasing range although this may be violated by quiet zones introduced by topographic features such as hills.

The number of targets within the detection range which can be simultaneously detected and isolated depends on factors such as array aperture, number of sensors in the array, noise level, signal level, and the azimuth separation of the targets. Lincoln's experience is that three to five would be an appropriate number. Equal power targets with a azimuth separation of less than 20 degrees may not be resolved.

The false alarm rates may depend on the targets present and the signal processing algorithm. In the absence of targets the number of false detections generated by the sensor and its associated signal processing algorithms is on the order

of three to five for each measurement interval.

The azimuth accuracy of the acoustic arrays is on the order of two degrees. This can be improved by changing the measurement intervals. A lower limit is imposed by propagation physics with a reasonable value of about one degree.

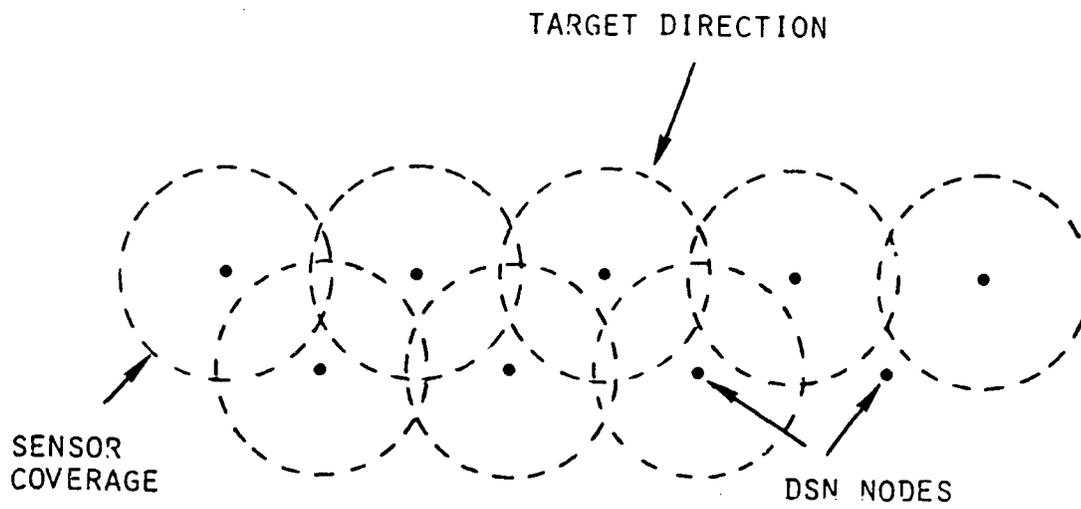
The sensors/processors can be deployed in various patterns. Two possible options are: barrier and area (Figure 3-1). The barrier pattern has the nodes arranged in a long linear extent and is useful for early warning situations. The area pattern has nodes in the interior of the DSN and is useful for continuous surveillance over large areas. A general DSN may contain many nodes but the system considered in the research has a maximum of six nodes so as to match the testbed hardware. The six nodes can be arranged in various ways to simulate the two options.

Important system deployment parameters are the distance between nodes, sensor detection range and the broadcast communication range. The system may exhibit different characteristics as a function of these parameters. It is not clear what kinds of parameters will be optimal. However, the sensor detection range should be at least equal to the distance between nodes to provide some overlapping coverage. The broadcast range should be at least the distance between nodes and possibly larger so that information can propagate faster in the system.

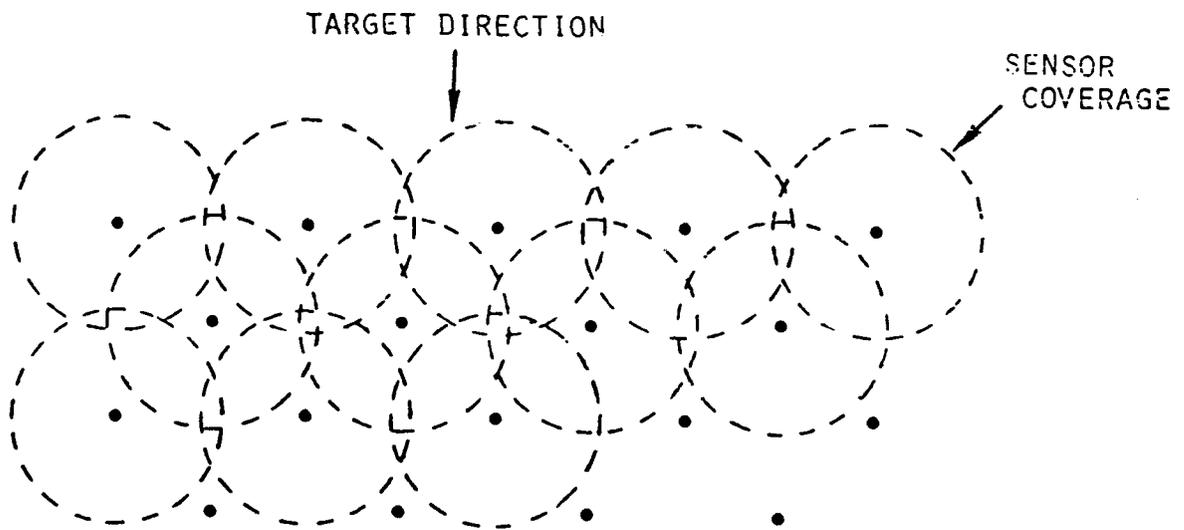
The nominal communication between nodes is a limited range unacknowledged broadcast. The nominal reception area is a disk with the range as the radius and the center at the broadcasting node. However, there may be "dead areas" within the disk where no reception is possible. These disks may be known or unknown to the system.

In addition, communication may be unreliable in the sense that messages could be lost. A message broadcast by one node may be received by some nodes within the broadcast disk and lost by others. This is equivalent to having failing communication channels. The broadcasting node may not be aware of the failure.

We are interested in target scenarios ranging from easy to quite difficult. A local target density of three per node would be considered quite difficult due to the limited resolution of the sensors. Thus, one, two, and three targets will be considered in our simulations. The targets may maneuver by changing courses or



(A) BARRIER



(B) AREA

Figure 3-1: DSN Deployment Patterns

speeds or both. They may pass through the DSN in various ways. The targets may fly in different configurations. For example, they may fly in different formations and the formations may cross each other.

### 3.2 MODELS

The altitudes of targets are assumed to be very low and the targets are modeled as objects moving in the 2-dimensional space. The motion is modeled by constant velocity or constant acceleration (or if necessary constant jerk). A maneuvering target is modeled by additional white noise excitation to the target dynamics.

Let a target position viewed from a sensor at time  $t$  be  $x(t)$ . The sound wave received at time  $t$  by the sensor has originated from the target at time  $t-\delta$ , where the time delay  $\delta$  is determined by

$$|x(t-\delta)| = c \delta \quad (3.1)$$

with  $c$  being the speed of the sound in the air (See Figure 3-2). Eqn. (3.1) has a unique solution  $\delta$  provided  $x(\cdot)$  is differentiable and  $|\dot{x}(t)| < c$  (subsonic). Eqn. (1) determines the acoustic azimuth (measured clockwise from the north)  $\phi$  of the target with respect to the sensor. The measured acoustic azimuth  $\phi_M$  contains measurement error as

$$\phi_M = \phi + w \quad (3.2)$$

where  $w$  is modeled by an independent zero-mean gaussian random variable (r.v.) whose variance is yet to be specified.

Let the sound pressure at the 1-meter distance from the target be  $s_0$ . Then the sound pressure measurement  $s_M$  at the sensor is

$$s_M = G \frac{s_0}{r^2} \quad (3.3)$$

where  $r$  is the acoustic range, i.e.,  $r = c \delta$ , and  $G$  is the sensor gain. To account for propagation irregularity and other random factors, either additive or

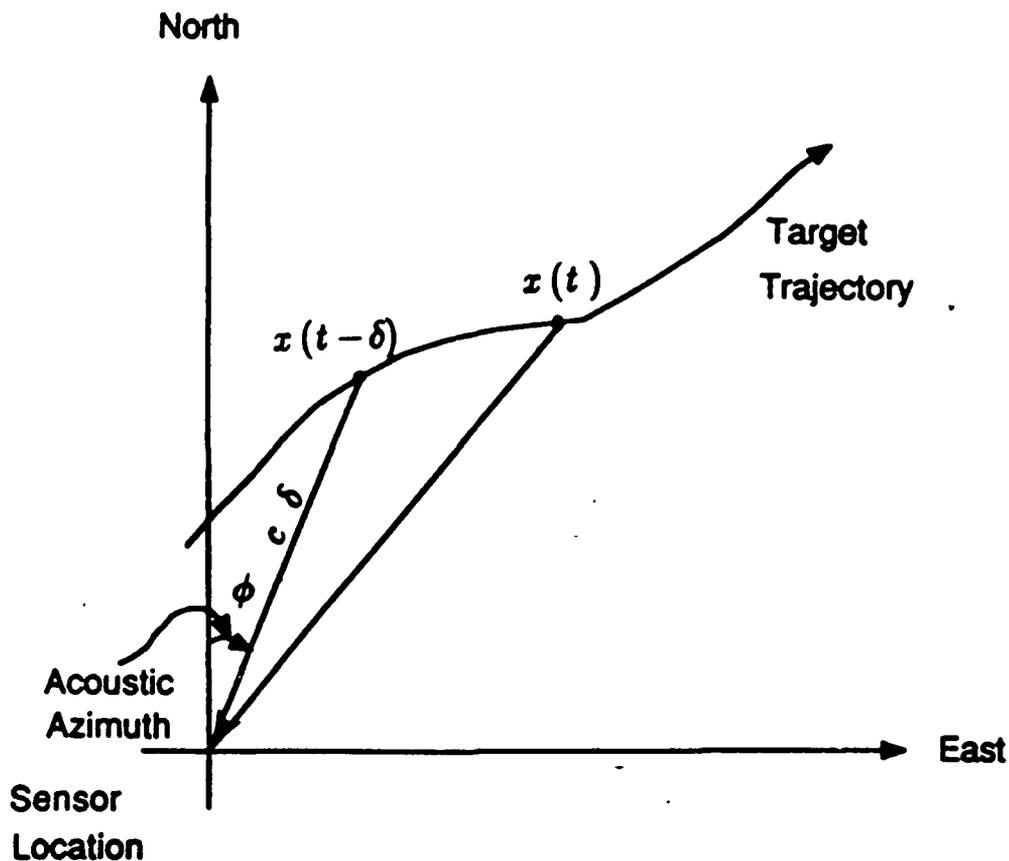


Figure 3-2: Target Sensor Geometry

multiplicative noise should be added to (3.3). The sensor also measures ambient noise  $s_N$ . Thus when the measured sound pressure exceeds a given threshold  $s_{TH}$ , the sensor generates a measurement  $y = (\phi_M, s_M, s_N)$  consisting of the acoustic azimuth and the signal/noise sound pressure levels.

The azimuth measurement error standard deviation (SD)  $\sigma_\phi$  (of  $w$  in (3.2)) is determined by

$$\sigma_\phi = \frac{\delta\phi}{\Gamma(SNR)} \quad (3.4)$$

where  $\delta\phi$  is the sensor resolution (about 20 degrees),  $SNR = s_M/s_N$ , and  $\Gamma(SNR) = \min\{\max\{1, \sqrt{SNR}\}, 10\}$ . The number of false alarms is modeled as a Poisson r.v. independent from scan to scan. The delayed azimuth value of a false alarm is distributed uniformly on  $[0, 2\pi]$  and the sound pressure value has an

exponential distribution biased by the threshold value.

When two acoustic azimuth measurements,  $\phi_M^1$  and  $\phi_M^2$ , are close enough, i.e.,  $|\phi_M^1 - \phi_M^2| < \delta\phi$ , they are merged into a single measurement, and the merged acoustic azimuth measurement becomes

$$\phi_M^m = q\phi_M^1 + (1-q)\phi_M^2 \quad (3.5)$$

where

$$q = \begin{cases} 1 & \text{if } s_M^1 > 5s_M^2 \\ \frac{s_M^1}{s_M^1 + s_M^2} & \text{otherwise} \end{cases} \quad (3.6)$$

and  $s_M^i$  is the unmerged sound pressure measurement corresponding to  $\phi_M^i$ . The merged sound pressure measurement becomes

$$s_M^m = \begin{cases} s_M^1 & \text{if } s_M^1 > 5s_M^2 \\ s_M^1 + \frac{1}{2}s_M^2 & \text{otherwise} \end{cases} \quad (3.7)$$

In Eqns. (3.5) to (3.7), we assume  $s_M^1 > s_M^2$ . Otherwise we should exchange indices 1 and 2.

## 4. LOCAL DATA PROCESSING

At each scan, a DSN processor receives azimuth and sound pressure measurements from its own acoustic sensor. This section discusses the processing of the local measurements based on the Generalized Tracker/Classifier (GTC) developed in [1]. The general approach is the same as that in Section 2. However, because of the nature of acoustic sensors, modifications to the general algorithm have to be made. Section 4.1 describes a modified representation of the tracks. Sections 4.2, 4.3, and 4.4 present the modifications made to hypothesis formation, evaluation, and track updating. Section 4.5 describes an approach used to handle wide variations in target dynamics. A summary of these results can be found in [13].

### 4.1 TRACK REPRESENTATION

As in [1],[2] or [3] the tracking data at each node is represented in terms of tracks and hypotheses. A *track* is a collection of *measurement indices*. For example, a track  $\tau = \{(1,2,3),(2,3,4),\dots\}$  hypothesizes that a target generates measurement 1 at scan 2 at sensor 3, measurement 2 at scan 3 at sensor 4, and so forth. The lack of measurement may be represented by a triple  $(0,k,s)$  for scan  $k$  of sensor  $s$  or simply excluded from the track. In general, two tracks may be inconsistent with each other, e.g., if one track is *true* the other one must be *false*. A consistent collection of tracks is called a *data-to-data association hypothesis* or simply a hypothesis. Each track  $\tau$  is accompanied by a *target state distribution (TSD)* which represents the distribution  $p(x_t | \tau, Z)$  of target state  $x_t$  conditioned by the track  $\tau$  and the accumulated sensor data  $Z$ . Since it is difficult for a node to generate estimates on position and velocity from the measurements of a single sensor, we distinguish between different kinds of tracks based on the target state distributions associated with them.

Each target state distribution (TSD) consists of a *geolocational TSD (GTSD) factor(s)* and a *sound pressure TSD (SPTSD) factor*. A TSD factor is called *local* if the corresponding distribution is derived from the measurements of a single sensor, and otherwise *global*. Thus a GTSD factor is either global or local while a SPTSD factor is always local. A GTSD or SPTSD factor consists

generally of multiple gaussian terms with a probabilistic weight being attached to each term. A track may have only a local GTSD factor. In such a case, a track is said to be *local*. Or a track may have a global GTSD factor or both global and local GTSD factors. Then the track is said to be *global*.

A local GTSD factor term is a gaussian distribution on the (local) acoustic azimuth of a target and its derivative,  $(\phi, \dot{\phi})$ , and possibly higher-order derivative(s). The purpose of having local GTSD factors is to overcome the difficulty of initiating tracks locally from acoustic azimuth measurements. As a local track accumulates acoustic azimuth data, the acoustic azimuth rate  $\dot{\phi}$  is estimated with increasing accuracy as indicated by the decreasing variance matrix in the local GTSD factor terms. A global GTSD factor term is a gaussian distribution on the global coordinates, i.e., the target position and the velocity in the north-east coordinate, and possibly their higher-order derivatives. A global GTSD factor term is generated by fusing two local tracks when different sensors communicate

A SPTSD factor tracks the change in the measured sound pressure. Its purpose is: 1) to obtain additional discriminant (particularly from false alarms), 2) to predict a target leaving the sensor coverage, and 3) to predict the merged acoustic azimuth measurements when measurement merging is likely. The factor is also used to estimate the targets' noisiness. A SPTSD factor term is a gaussian distribution on the (fictitiously noiseless) received sound pressure  $s$ , its derivative, and possibly higher-order derivative(s). The actually measured sound pressure  $s_M$  is modeled by

$$s_M = s + w_s \quad (4.1)$$

where the artificial noise term  $w_s$  (modeled by independent zero-mean gaussian r.v.) accounts for scan-to-scan fluctuation of the sound pressure measurements. Figure 4-1 shows the hierarchy in track representation performed.

The updating of each TSD factor is in parallel to the hypothesis evaluation (described in Section 4.4). On the other hand, extrapolation of each TSD factor term is performed by an appropriate dynamic model, i.e., constant-velocity or constant-acceleration linear models with an appropriate white noise input. For example, in order to update a local GTSD factor term, a simple set of differential

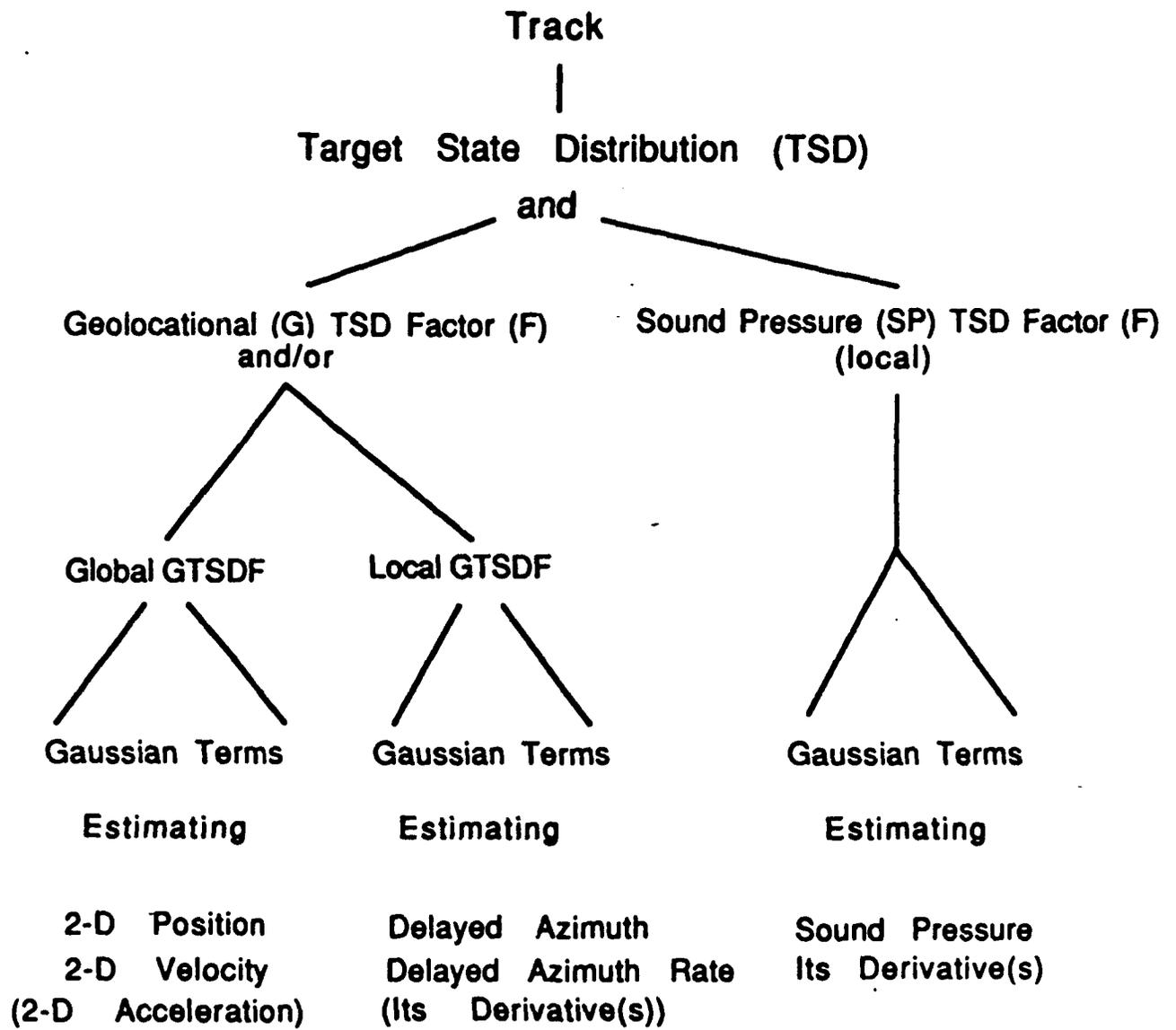


Figure 4-1: Track Representation

equations,  $\frac{d}{dt}\hat{\phi}=\dot{\hat{\phi}}$  and  $\frac{d}{dt}\hat{\phi}=\text{white noise}$ , may be used. Since the acoustic azimuth dynamics are in fact nonlinear, the intensity of white noise must be chosen to compensate for such nonlinearity in addition to the target maneuvering or multiple models (described in Section 4.5) have to be used.

## 4.2 HYPOTHESIS FORMATION

As described in [1, 2, 3], hypotheses are recursively created, i.e., expanded and evaluated for each data set from each sensor. To simplify our discussion, we consider only two-way measurement merging as modeled in Section 3, i.e., only the possibility of two existing tracks merging together will be considered. For the sake of efficiency, hypotheses are usually clustered [4, 5] in the processing. In the following, we discuss hypothesis formation and evaluation within a single cluster.

Consider one sensor scan and a hypothesis  $\bar{\lambda}$  to be expanded. Then, before expanding this hypothesis by the set of measurements in the data set, it is expanded into the set of *track merging hypotheses*, each of which is a partition  $\bar{\Lambda}_m$  of  $\bar{\lambda}$  such that  $\#(\bar{\mathbf{T}}) \leq 2$  for any  $\bar{\mathbf{T}} \in \bar{\Lambda}_m$ . Here  $\#(A)$  is the number of members in a set  $A$ . Apparently, this expansion means that we are only considering two-way merging among existing tracks. Then each track merging hypothesis is expanded by the set of measurements as in the cases where there is no measurement merging. Figure 4-2 illustrates this two-step hypothesis expansion; first by track merging and next by the measurements. In the figure, a hypothesis  $\bar{\lambda}$  having three tracks is expanded into four track merging hypotheses,  $\bar{\Lambda}_m^1$  to  $\bar{\Lambda}_m^4$ , each of which is then expanded by the measurements (shown by shaded triangles in Figure 4-2). Figure 4-3 shows the expansion of the hypothesis  $\bar{\Lambda}_m^1$  by the two measurements in the current sensor scan.

## 4.3 HYPOTHESIS EVALUATION

After expanding all the hypotheses  $\bar{\lambda}$  in the old cluster, the resultant collection of new hypotheses forms an updated cluster. Each new hypothesis  $\lambda$  has a unique parent  $\bar{\lambda}$  and a unique track merging hypothesis  $\bar{\Lambda}_m$ , from which  $\lambda$  is generated. Then evaluation of hypotheses, considering the measurement merging possibility, can be done by replacing  $\bar{\lambda}$  by  $\bar{\Lambda}_m$  in the general hypothesis evaluation formula given in [2] and [6], and then by probabilistically assessing the joint event

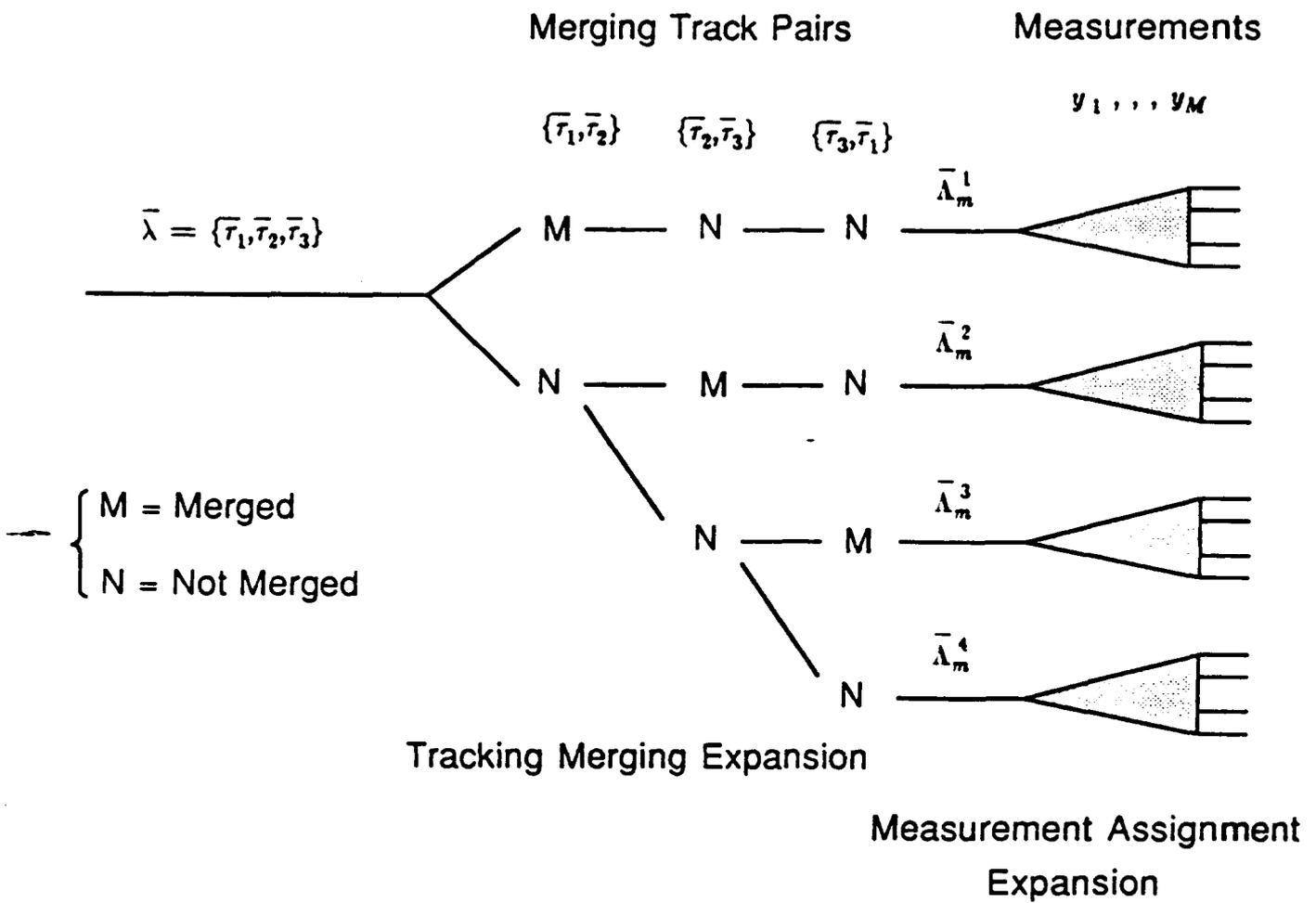


Figure 4-2: Hypothesis Expansion with Measurement Merging Possibility (1)

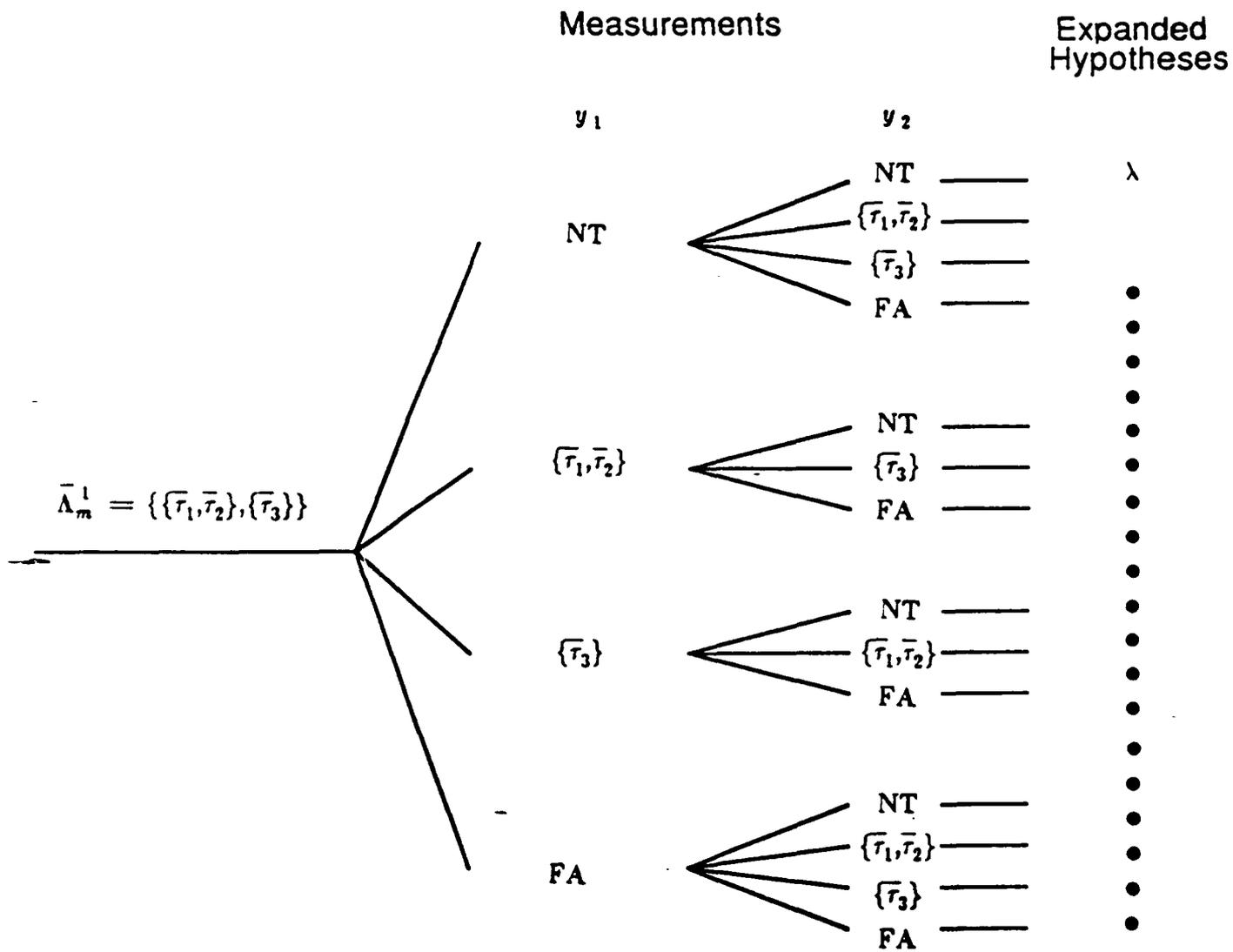


Figure 4-3: Hypothesis Expansion with Measurement Merging Possibility (2)

of "two tracks merged and generating a single measurement."

The results of hypothesis evaluation may be summarized as

$$\begin{aligned}
 Prob. \{ \lambda | Z \} &= C^{-1} Prob. \{ \bar{\lambda} | \bar{Z} \} \\
 &\left( \prod \{ L(y | \bar{\mathbf{T}}) | \bar{\mathbf{T}} \in \bar{\Lambda}_m \cup \{ \emptyset \} \text{ and } \bar{\mathbf{T}} \text{ is assigned measurement } y \} \right) \\
 &\left( \prod \{ L(\theta | \{ \bar{\tau}_1, \bar{\tau}_2 \}) | \tau_1 \in \lambda \text{ and } \tau_2 \in \lambda \text{ but they did not merge} \} \right) \\
 &\left( \prod \{ L(\theta | \bar{\tau}) | \tau \in \lambda \text{ but not assigned any measurement, i.e., } \bar{\tau} = \tau \} \right) \quad (4.2)
 \end{aligned}$$

where  $Z$  is the cumulative data set including the current sensor scan,  $\bar{Z}$  is  $Z$  minus the current sensor scan, and  $C$  is the normalizing constant.  $\theta$  in the above equation is the symbol used to represent "no measurement" and the  $L(\cdot | \cdot)$ 's are likelihood functions defined below.

In the first parenthesized product on the right hand side of (4.2),  $L(y | \emptyset)$  is the likelihood of measurement  $y$  originating from a target undetected before and is given by

$$L(y | \emptyset) = \beta_{NT}(\phi_M) / \beta_{FA} \quad (4.3)$$

where  $\beta_{NT}(\cdot)$  is the expected density of undetected targets, translated into the acoustic azimuth space, i.e.,  $[0, 2\pi]$ , and  $\beta_{FA} = \nu_{FA} / 2\pi$  is the density of the false alarms over the  $[0, 2\pi]$  interval, where  $\nu_{FA}$  is the expected number of false alarms (about from 1 to 3) per scan. Eqn. (4.3) also assumes that the sound pressure measurement distribution of a target "heard" (detected) for the first time is equal to that of a false alarm.

When  $\bar{\mathbf{T}} \neq \emptyset$ ,  $L(y | \bar{\mathbf{T}})$  is the likelihood of measurement  $y$  originating from an existing track  $\bar{\mathbf{T}} = \{ \bar{\tau} \}$  or jointly from two existing tracks  $\bar{\mathbf{T}} = \{ \bar{\tau}_1, \bar{\tau}_2 \}$  and is defined by

$$L(y | \bar{\mathbf{T}}) = \frac{L_\phi(\phi_M | \bar{\mathbf{T}})}{\beta_{FA}} \frac{L_s(s_M | \bar{\mathbf{T}})}{p_s^{FA}(s_M)} \quad (4.4)$$

where  $p_s^{FA}$  is the probability density of the false alarm sound pressure. For simplicity, we assume that the GTSD and SPTSD factors of each track are both single-termed. The extension to multiple term TSD factor is rather straightforward and will be explained later. In case  $\#(\bar{\mathbf{T}})=1$ , i.e., when there is no merging, we have

$$L_\phi(\phi_M | \{\bar{\tau}\}) = g(\phi_M - \bar{\phi}; \tilde{\sigma}_\phi) \quad (4.5)$$

where  $g(\xi; \sigma) \triangleq \exp(-\xi^2/2)/(\sqrt{2\pi}\sigma)$  is the probability density of a zero-mean gaussian variable,  $\bar{\phi}$  is the acoustic azimuth prediction by a local or global GTSD factor of  $\bar{\tau}$ ,  $\tilde{\sigma}_\phi^2$  is the corresponding innovations variance, and

$$L_s(s_M | \{\bar{\tau}\}) = g(s_M - \bar{s}; \tilde{\sigma}_s) \quad (4.6)$$

with  $\bar{s}$  being the sound pressure prediction by the (local) SPTSD factor of  $\bar{\tau}$  and  $\tilde{\sigma}_s^2$  being the corresponding innovations variance. Equations (4.5) and (4.6) are likelihood functions commonly used in many multitasking algorithms.

In order to calculate the likelihood function when  $\#(\bar{\mathbf{T}})=2$ , i.e., when measurement merging occurs, we must make some approximations. First we approximate  $q$  in (3.5) by

$$\bar{q} = \begin{cases} 1 & \text{if } \bar{s}_1 > 5\bar{s}_2 \\ \frac{\bar{s}_1}{\bar{s}_1 + \bar{s}_2} & \text{otherwise} \end{cases} \quad (4.7)$$

where  $\bar{s}_1$  and  $\bar{s}_2$  are the sound pressure predictions of the two tracks. Likewise, we approximate (3.7) by

$$s_M^m = \begin{cases} s_M^1 & \text{if } \bar{s}_1 > 5\bar{s}_2 \\ s_M^1 + \frac{1}{2}s_M^2 & \text{otherwise} \end{cases} \quad (4.8)$$

We denote the right hand side of (4.8) as  $h_s^m(s_M^1, s_M^2; \bar{s}_1, \bar{s}_2)$ . With these approximations, we have

$$L_{\phi}(\phi_M | \{\bar{\tau}_1, \bar{\tau}_2\}) = g(\phi_M - h_{\phi}^m(\bar{\phi}_1, \bar{\phi}_2; \bar{q}); \tilde{\sigma}_{\phi}^m) \left[ \operatorname{erf}\left(\frac{\delta\phi - \tilde{\Delta}\phi}{\tilde{\sigma}_{\Delta\phi}}\right) - \operatorname{erf}\left(\frac{-\delta\phi - \tilde{\Delta}\phi}{\tilde{\sigma}_{\Delta\phi}}\right) \right] \quad (4.9)$$

where  $h_{\phi}^m(\bar{\phi}_1, \bar{\phi}_2; \bar{q}) \triangleq \bar{q}\bar{\phi}_1 + (1 - \bar{q})\bar{\phi}_2$ ,  $\operatorname{erf}(x) \triangleq \int_{-\infty}^x g(\xi) d\xi$  is the error function, and

$$\tilde{\sigma}_{\phi}^m = \sqrt{[\bar{q}\sigma_w^1]^2 + [(1 - \bar{q})\sigma_w^2]^2 + [\bar{\sigma}_{\phi}^1]^2 + [\bar{\sigma}_{\phi}^2]^2} \quad (4.10)$$

with  $\sigma_w^i$  being the SD of the measurement noise in (3.2), and  $\bar{\sigma}_{\phi}^i$  being the SD of the acoustic azimuth prediction error determined by  $\bar{\tau}_i$  for each  $i$ . The other parameters are

$$\tilde{\Delta}\phi = \bar{\phi}_1 - \bar{\phi}_2 + \frac{\bar{q}\tilde{P}_1 - (1 - \bar{q})\tilde{P}_2}{\bar{q}^2\tilde{P}_1 + (1 - \bar{q})^2\tilde{P}_2} (\phi_M - h_{\phi}^m(\bar{\phi}_1, \bar{\phi}_2; \bar{q})) \quad (4.11)$$

and

$$\tilde{\sigma}_{\Delta\phi} = \sqrt{\frac{P_1 P_2}{\bar{q}^2\tilde{P}_1 + (1 - \bar{q})^2\tilde{P}_2}} \quad (4.12)$$

where  $\bar{\phi}_i$  is the acoustic azimuth prediction by the local or global GTSD factor of track  $\bar{\tau}_i$  and  $\tilde{P}_i = [\bar{\sigma}_{\phi}^i]^2 + [\sigma_w^i]^2$ , for each  $i$ .

For the sound pressure part, we have

$$L_s(s_M | \{\bar{\tau}_1, \bar{\tau}_2\}) = g(s_M - h_s^m(\bar{s}_1, \bar{s}_2; \bar{s}_1, \bar{s}_2); \tilde{\sigma}_s^m(\bar{s}_1, \bar{s}_2)) \quad (4.13)$$

where

$$\tilde{\sigma}_s^m(\bar{s}_1, \bar{s}_2) = \sqrt{[\sigma_s^m(\bar{s}_1, \bar{s}_2)]^2 + [\bar{\sigma}_s^1]^2 + [\bar{\sigma}_s^2]^2} \quad (4.14)$$

$\bar{\sigma}_s^1$  is the SD of the sound pressure prediction error determined by the SPTSD factor of track  $\bar{\tau}_i$  for each  $i$ , and  $\sigma_s^m(\bar{s}_1, \bar{s}_2)$  is either  $\sqrt{5/2}\sigma_w^s$  or  $\sigma_w^s$ , depending on the condition in (4.8), with  $\sigma_w^s$  being the SD of the noise term in (3.8). The derivations of (4.9) and (4.13) are described in the Appendix A.

$L(\theta | \{\bar{\tau}_1, \bar{\tau}_2\})$  is the likelihood (probability) of tracks,  $\bar{\tau}_1$  and  $\bar{\tau}_2$ , not being merged, i.e.,

$$L(\theta | \{\bar{\tau}_1, \bar{\tau}_2\}) = 1 - Prob. \{ \{\bar{\tau}_1, \bar{\tau}_2\} \} \quad (4.15)$$

with

$$Prob. \{ \{\bar{\tau}_1, \bar{\tau}_2\} \} = \operatorname{erf}\left(\frac{\delta\phi - \overline{\Delta\phi}}{\sqrt{\bar{P}_1 + \bar{P}_2}}\right) - \operatorname{erf}\left(\frac{-\delta\phi - \overline{\Delta\phi}}{\sqrt{\bar{P}_1 + \bar{P}_2}}\right) \quad (4.16)$$

being the probability of the two tracks being merged, where  $\overline{\Delta\phi} = \bar{\phi}_1 - \bar{\phi}_2$ .

The target detection model yields the likelihood (probability) of a target hypothesized by the track  $\bar{\tau}$  being undetected in the current scan, i.e.,

$$L(\theta | \{\bar{\tau}\}) = 1 - \operatorname{erf}\left(\frac{s_{TH} - s}{\bar{\sigma}_s}\right) \quad (4.17)$$

Thus the evaluation of the newly expanded hypotheses is equivalent to the calculation of all the likelihood functions defined above. Therefore, it is convenient to store all the above likelihoods in a table. We call such a table an *extended (because it includes merged measurements) track-to-measurement cross-reference table*.

#### 4.4 TRACK UPDATE

Parallel to the calculation of each likelihood, we update each track according to the assumed measurement assignment. When a measurement is assigned to a single track, both the GTSD and SPTSD factors of the track are updated by the Kalman filter or the extended Kalman filter.

A track containing a global GTSD factor is called a global track. A global track is obtained either by fusing two local tracks together or from alert/handover information sent by other nodes. In updating the global GTSD factors using the local azimuth measurements, an extended Kalman filter is needed because of the non-linear measurement model. For each global GTSD factor, the likelihood function (4.5) can be obtained as,

$$L_{\phi}(\phi_M | \{\bar{\tau}\}) = g(\phi_M - \bar{\phi}_S ; \tilde{\sigma}_{\phi_S}) \quad (4.18)$$

where  $\bar{\phi}_S$  is the acoustic azimuth prediction by a global GTSD factor of  $\bar{\tau}$ ,  $\tilde{\sigma}_{\phi_S}^2$  is the corresponding innovations variance and is obtained as

$$\tilde{\sigma}_{\phi_S}^2 = H\bar{\Sigma}H^T + \sigma_{\phi}^2 \quad (4.19)$$

In Equation (4.19),  $\bar{\Sigma}$  is the covariance matrix of the global GTSD factor,  $H$  is the partial derivatives of the azimuth respective to the global states given by

$$\begin{aligned} H &= \left[ \frac{\partial \phi}{\partial u_N}, \frac{\partial \phi}{\partial u_E}, \frac{\partial \phi}{\partial v_N}, \frac{\partial \phi}{\partial v_E} \right] \\ &= \left[ -\frac{M}{DR^2} u_E, \frac{M}{DR^2} u_N, \frac{1}{DcR} u_E, -\frac{1}{DcR} u_N \right] \end{aligned} \quad (4.20)$$

$c$  is the speed of sound,  $D$  is given by

$$D = M^{-1} (1 + \xi \cos \beta) \quad (4.21)$$

where

$$M = \sqrt{1 + \xi^2 + 2\xi \cos \beta} \quad (4.22)$$

is the ratio of actual target range  $R$  to the acoustic range  $r$  with  $\xi \triangleq V/c$  being the Mach number,  $\beta \triangleq \phi - \psi$  being the acoustic aspect angle, and  $\psi$  being the target heading. A summary of the geometry and relevant notations for two-dimensional subsonic acoustic tracking can be found in Appendix B.

With the partial derivatives obtained as above, the updated global GTSD factor mean and covariance can be calculated using the standard Kalman filter equations. Namely,

$$\hat{x} = \bar{x} + K(\phi_M - \bar{\phi}_S) \quad (4.23)$$

$$K = \bar{\Sigma} H^T S^{-1} \quad (4.24)$$

$$S = H \bar{\Sigma} H^T + \sigma_v^2 \quad (4.25)$$

and

$$\bar{\Sigma} = (I - KH) \bar{\Sigma} \quad (4.26)$$

When a measurement is assigned to two merged tracks, using the approximate joint measurement equations, eqn. (3.5) with  $q$  replaced by  $\bar{q}$  (for the GTSD factor) and eqn. (4.7) (for the SPTSD factor), the GTSD and SPTSD factors can be jointly updated. The resulting cross-correlation between the two tracks is then ignored for simplicity. When no measurement is assigned to a track, the TSD factors are not updated. When a measurement is assigned to the null track, i.e., a single measurement is used to initiate a new track, a single-term local GTSD factor and a SPTSD factor are generated using the appropriate variance matrices.

Updated clusters are then subject to *hypothesis management* operations including 1) hypothesis pruning in which low-probability hypotheses are cut off, 2) hypothesis combining in which similar hypotheses are combined, and 3) cluster splitting in which confirmed or nearly confirmed tracks are split from a cluster.

#### 4.5 MULTIPLE MODEL APPROACH

In general, a single term TSD factor may not be sufficient for a local track. This is because widely different target dynamics may have to be considered in tracking. For instance, when tracking a target with varying speeds, one must allow for large changes in the azimuth rate relative to a sensor. Different process noises are then needed to model the target at different times. Similar situations happen in tracking multiple targets with different speeds where it becomes inappropriate to model all targets with a process noise of a single intensity.

One way to handle this is by means of so called *Multiple Model* (MM) approach. In the MM approach, several models with different dynamics or parameters are used. Starting with equal or pre-determined probabilities (weights) for each model when the track was first initiated the measurements at each sampling

interval are used to update each model and calculate the association likelihoods. The new model probabilities can then be obtained by considering both the prior model probabilities and measurement likelihoods.

$$\{m_i(\tau)\} = \frac{1}{C} P\{m_i(\bar{\tau})\} L_\delta(\phi_M | \bar{r}, m_i(\bar{\tau})) \quad (4.27)$$

where  $C$  is the normalization constant,  $P\{m_i(\tau)\}$  and  $P\{m_i(\bar{\tau})\}$  are the new and prior model probabilities, and  $L_\delta$  is the association likelihood.

With different models for slow and fast target, when tracking the slow constant speed target the slow model will dominate and vice versa. While tracking a maneuvering target, the model probability will shift between slow and fast models depending on the target speed. With this approach, if a sufficient number of target models are given, one needs not know the actual target speed *a priori*. The algorithm will converge to the right model automatically. Some numerical examples will be given in Section 6 to demonstrate the feasibility of this concept.

## 5. INFORMATION FUSION AND DISTRIBUTION

Communication among processing nodes allows each node to use information that is otherwise not available locally. This section discusses the algorithms for information fusion and distribution. Although the general framework of Section 2 is applicable, again we have to modify the algorithms for acoustic sensors. Section 5.1 describes the information fusion algorithms and Section 5.2 discusses communication strategies. A summary of the results can be found in [14] and [15].

### 5.1 FUSION PROCESSING

When hypotheses are received from another node, they are fused with the hypotheses at the node to form new hypotheses. As in local data processing, the basic steps include hypothesis formation, evaluation and management. The distributed nature of the processing necessitates operations for checking that only consistent hypotheses are formed and removing redundant information in hypothesis evaluation. These operations are facilitated by means of the information graph [2],[7] which is an abstract model of the communication and processing in the DSN. In the following discussion, we use the terms *home* and *foreign* to represent the information present in the local node and that coming in from an external node, respectively.

Although we discuss hypothesis formation and evaluation separately, in actual implementation they are usually performed simultaneously so that no unnecessary hypothesis expansion is used. For example, it is possible that a hypothesis pair  $(\bar{\lambda}_1, \bar{\lambda}_2)$  satisfies the necessary condition for fusability but yields zero probability upon evaluation. In such a case, the hypotheses need not be fused at all.

### 5.1.1 Hypothesis Formation

The goal of hypothesis formation is to generate hypotheses on home and foreign tracks which come from the same targets. In a distributed system the key problem is identifying the fusable hypotheses and tracks from the home and foreign hypotheses and tracks. Two hypotheses are not fusable if they came from mutually exclusive hypotheses at an earlier time. Similarly, two tracks are not fusable if they came from disjoint tracks. According to [2],[7], the entire fusion problem can be defined in terms of the information graph. Both the home and foreign information states (tracks and hypotheses, etc.) are defined at information nodes  $i_1$  and  $i_2$  in the information graph. Then consistency checking for hypothesis formation starts by finding the *minimum set* of common predecessors of  $i_1$  and  $i_2$  in the information graph. By tracing back the graph to this minimum set, fusability can be determined.

A home track  $\bar{\tau}_1$  and a foreign track  $\bar{\tau}_2$  are fused whenever they are fusable. The two tracks are fusable if and only if they share the same predecessor track on each information node in the common predecessor set.

### 5.1.2 Hypothesis Evaluation

Each fused hypothesis is evaluated using the following equation. Let  $\lambda$  be a fused hypothesis and  $Z$  be the cumulative data at the fusion information node, then

$$P(\lambda | Z) = C^{-1} \prod_{i \in I_R} P(\lambda_{|i} | Z_i)^{\alpha(i)} \prod_{\tau \in \lambda} l(\tau) \quad (5.1)$$

where  $C$  is the normalizing constant,  $(I_R, \alpha)$  is the *information redundancy indicator*,  $\lambda_{|i}$  is the predecessor of  $\lambda$  on the information node  $i$ , and  $l(\tau) = L(\bar{\tau}_1, \bar{\tau}_2)$  with  $(\bar{\tau}_1, \bar{\tau}_2)$  being the pair of tracks uniquely determined by a fused track  $\tau$ .  $(I_R, \alpha)$  represents the redundant information at the two information nodes  $i_1$  and  $i_2$ .  $I_R$  is the set of information nodes which affect the common information and  $\alpha$  is an integer-valued function which indicates how the information should be handled. For example, if  $I_R = \{i_1, i_2, i_3\}$ , then  $\alpha(i_1) = 1$ ,  $\alpha(i_2) = 1$  and  $\alpha(i_3) = -1$  means that

when the information at  $i_1$  and  $i_2$  is fused, the redundant information at  $i_3$  has to be removed.

A key step in hypothesis evaluation is the computation of the track-to-track likelihood  $L(\bar{\tau}_1, \bar{\tau}_2)$  for every fusible pair  $(\bar{\tau}_1, \bar{\tau}_2)$  of home and foreign tracks. For each of the tracks in the given pair, the last time when the track was updated is examined. If the updating times are different, the TSD of the track which has not been recently updated is extrapolated so that the two TSD's correspond to the target state at the same time. Then the track-to-track likelihood is calculated from the GTSD factors of the two tracks.

Whenever the likelihood is positive, the fused track  $\tau = \bar{\tau}_1 \cup \bar{\tau}_2$  is created. Each fused track  $\tau$  is then associated with a fused TSD (target state distribution) which is created by fusing the TSD's of the tracks from which it is formed. The GTSD factor for the fused track is created from the GTSD factors of the tracks from which it is fused. The SPTSD factor of the fused track is the same as that of the home track in the track pair to be fused.

Since the home and foreign tracks may be local or global or even empty, the computation of the track-to-track likelihoods has to consider all these possibilities. The different types of track-to-track likelihoods are shown in Figure 5-1. The calculation of the track-to-track likelihood and the fused GTSD factor for each fused track is described in the following for all possible combinations of home and foreign GTSD factor. Because of symmetry, some of the combinations can be omitted. To simplify the notation, we assume that each GTSD factor only has a single term. The results can be generalized to cases involving sum-of-gaussians.

### CASE 1: Local Home/Local Foreign

This is the case when two local tracks from two sensor nodes are fused to initiate a global track. When the home track  $\tau_1$  and foreign track  $\tau_2$  are both local  $(\tau_1 \cup \tau_2)|_i = \emptyset$  except for  $i = i_1$  and  $i = i_2$ . The fused GTSD factor is created first. This is done by using the "position" track initiation equation described in [8, 9]. As before, we assume that both home and foreign tracks have single-termed GTSD factors. Then, using the estimates of the two local azimuth values and their first-order derivatives,  $\Phi \triangleq [\phi_1, \phi_2, \dot{\phi}_1, \dot{\phi}_2]^T$ , a quadratic equation for the global state  $x \triangleq [u_N, u_E, v_N, v_E]^T$ , can be derived :

		HOME TRACKS		
		LOCAL	GLOBAL	EMPTY
FOREIGN TRACKS	LOCAL	1	2	5
	GLOBAL	2	3	6
	EMPTY	4	4	7

Figure 5-1: Possible Track-to-Track Combinations

where  $(\hat{V}_k, \hat{i}_k)$  is the target speed and heading estimated by  $\hat{x}_k$ ,  $\bar{p}^V(\cdot, \cdot)$  is the density of the *a priori* distribution of the target velocity vector, and  $J_i$  is the Jacobian associated with the global-to-local coordinate transformation for each sensor  $i$ .

Then the GTSD factor of the fused track becomes a two-term sum-of-gaussian distribution with probability weights  $w_1$  and  $w_2$  given by

$$\frac{w_1}{w_2} = \frac{c_1}{c_2} \quad (5.7)$$

and

$$w_1 + w_2 = 1 \quad (5.8)$$

The track-to-track likelihood is calculated as

$$L(\tau_1, \tau_2) = (\bar{\beta}_u)^{-1} (\det(\Sigma_{\phi_1}))^{-\frac{1}{2}} (\det(\Sigma_{\phi_2}))^{-\frac{1}{2}} (c_1 + c_2) \quad (5.9)$$

where  $\Sigma_{\phi_1}$  and  $\Sigma_{\phi_2}$  are the variance matrices of the local (azimuth and its derivative) estimates, and  $\bar{\beta}_u$  is the density of undetected targets per unit area.

In many cases, we have either  $w_1 \ll w_2$  or  $w_2 \ll w_1$ , indicating that only one solution to the algebraic equation is valid and the other corresponds to an unlikely estimate. Thus, except for rare occasions, only a single-term global GTSD factor is needed.

### CASE 2: Global Home/Local Foreign

When the foreign track  $\tau_2$  is local, every predecessor track  $(\tau_1 \cup \tau_2)|_i$  is empty for every  $i \in I_R$  except for  $i_1$  and  $i_2$ . If the foreign track has a global GTSD, the calculation of the likelihood and the fused GTSD can be done as in CASE 3.

Suppose the foreign track has a local GTSD factor. Then the GTSD factor of the fused track  $\tau_1 \cup \tau_2$  has the mean  $\hat{x}$  and variance matrix  $\Sigma$ , which are calculated by the extended Kalman filter equations:

$$\hat{x} = \hat{x}_1 + K(\hat{\Phi}_2 - \bar{\Phi}_1) \quad (5.10)$$

where  $\hat{x}_1$  is the mean of the GTSD of the home track  $\tau_1$ ,  $\hat{\Phi}_2$  is the vector of the means of the acoustic azimuth and its derivative in the GTSD factor of the foreign track  $\tau_2$ ,  $\bar{\Phi}_1$  is the azimuth and its derivative of the target as predicted by  $\hat{x}_1$ .

$K$  in (5.10) is the filter gain defined by

$$K = \bar{\Sigma}_1 H^T S^{-1} \quad (5.11)$$

where

$$S = H \bar{\Sigma}_1 H^T + R \quad (5.12)$$

$\bar{\Sigma}_1$  is the variance matrix of the GTSD factor of the home track  $\tau_1$ ,  $R$  is the variance submatrix of the local GTSD factor of the foreign track  $\tau_2$ , and  $H$  is the derivative of the transformation function  $h$  which transforms the global target state into the local coordinates used for the GTSD component of the foreign track  $\tau_2$ , and is given by

$$H = \begin{bmatrix} \frac{\partial \phi}{\partial u_N} & \frac{\partial \phi}{\partial u_E} & \frac{\partial \phi}{\partial v_N} & \frac{\partial \phi}{\partial v_E} \\ \frac{\partial \dot{\phi}}{\partial u_N} & \frac{\partial \dot{\phi}}{\partial u_E} & \frac{\partial \dot{\phi}}{\partial v_N} & \frac{\partial \dot{\phi}}{\partial v_E} \end{bmatrix} \quad (5.13)$$

where the first row of  $H$  has been obtained in (4.20) and the second row are obtained as,

$$\frac{\partial \dot{\phi}}{\partial u_N} = \frac{Mc}{D^3 R^2} (D \xi \sin \alpha \cos \phi + N \sin \theta) \quad (5.14)$$

$$\frac{\partial \dot{\phi}}{\partial u_E} = \frac{Mc}{D^3 R^2} (D \xi \sin \alpha \sin \phi - N \cos \theta) \quad (5.15)$$

$$\frac{\partial \dot{\phi}}{\partial v_N} = \frac{1}{D^3 R} (-D \xi \sin \alpha \cos \phi - M \sin \theta) \quad (5.16)$$

$$\frac{\partial \dot{\phi}}{\partial v_E} = \frac{1}{D^3 R} (-D \xi \sin \alpha \sin \phi - M \cos \theta) \quad (5.17)$$

In the above,  $N = M - D$ ,  $\theta$  is the current target azimuth,  $\alpha \triangleq \theta - \psi$  is the target aspect angle, and the remaining notations have been defined in Section 4 and Appendix B.

The variance  $\Sigma$  of the fused track is then given by

$$\Sigma = (I - KH) \bar{\Sigma}_1 \quad (5.18)$$

The track-to-track likelihood is calculated as

$$L(\tau_1, \tau_2) = \bar{\beta}_u^{-1} (2\pi)^{-1} (\det(S))^{-\frac{1}{2}} \exp\left(-\frac{1}{2} |\hat{\Phi}_2 - \bar{\Phi}_1|_{s-1} \right) |J_2(\hat{x})| \quad (5.19)$$

where  $J_2(\cdot)$  is the Jacobian associated with the global-to-local coordinate transformation for the foreign track and is obtained as,

$$J_2(\cdot) = \det \left( \frac{\partial(\phi, \dot{\phi}, V, \psi)}{\partial(u, v)} \right) = -\frac{M^2}{D^2 R^3} \sin \alpha \quad (5.20)$$

### CASE 3: Global Home/Global Foreign

Suppose both the home and foreign tracks  $\tau_1$  and  $\tau_2$  are global. The information nodes in the set  $I_R$  belong to two classes: those where the common predecessor of  $\tau_1$  and  $\tau_2$  have a global GTSD factor and those where the common

predecessor of  $\tau_1$  and  $\tau_2$  have a local GTSD factor. Let

$$I_R^G = \{i \in I_R \mid (\tau_1 \cup \tau_2)|_i \text{ has a global GTSD factor}\} \quad (5.21)$$

Let  $\hat{x}_i$  and  $\Sigma_i$  be the mean and the variance of the global GTSD factor of the predecessor track  $(\tau_1 \cup \tau_2)|_i$  of the fused track  $(\tau_1 \cup \tau_2)$  at the node  $i \in I_R^G$ . Then the part of the track-to-track likelihood related to  $I_R^G$  is given by

$$L_G(\tau_1, \tau_2) = \left( \frac{\det(\Sigma)}{\prod_{i \in I_R^G} \det(\Sigma_i)^{\alpha_i}} \right)^{\frac{1}{2}} \exp\left(-\frac{1}{2} \sum_{i \in I_R^G} \alpha_i \|\hat{x} - \hat{x}_i\|_{\Sigma_i^{-1}}^2\right) \quad (5.22)$$

where  $\hat{x}$  is the global GTSD factor of the fused track  $\tau_1 \cup \tau_2$  and given by

$$\hat{x} = \Sigma \sum_{i \in I_R^G} \alpha(i) \Sigma_i^{-1} x_i \quad (5.23)$$

and  $\Sigma$  is the corresponding variance given by

$$\Sigma = \left( \sum_{i \in I_R^G} \alpha(i) \Sigma_i^{-1} \right)^{-1} \quad (5.24)$$

Note that equations (5.23) and (5.24) have the usual sum and difference terms to ensure that information is not used redundantly.

Let  $I_R^L = I_R \setminus I_R^G$ , i.e., the set of common predecessor nodes where the tracks have local GTSD factors. For each  $i$  in  $I_R^L$ , define

$$p_i(\tau_1, \tau_2) = p_i^L(\hat{\phi}, \hat{\phi}) \bar{p}^V(\hat{V}, \hat{\psi}) |J(\hat{x})| \quad (5.25)$$

where  $(\hat{\phi}, \hat{\phi})$  is the pair of the estimates of the acoustic azimuth and its derivative calculated from  $\hat{x}$ ,  $(\hat{V}, \hat{\psi})$  is the target speed/heading estimated by  $\hat{x}$ ,  $p_i^L(\cdot, \cdot)$  is the density of the GTSD factor of the predecessor track at  $i$  (marginal to  $(\phi, \dot{\phi})$ ),

$\bar{p}^V(\cdot, \cdot)$  is the density of the *a priori* distribution of the target speed and the heading, and  $J(\cdot)$  is the appropriate Jacobian (see Appendix B).

Define

$$L_L(\tau_1, \tau_2) = \prod_{i \in I_R^k} p_i(\tau_1, \tau_2)^{a(i)} \quad (5.26)$$

Then the track-to-track likelihood is calculated as

$$L(\tau_1, \tau_2) = \bar{\beta}_u^{-1} L_L(\tau_1, \tau_2) L_G(\tau_1, \tau_2) \quad (5.27)$$

where  $\bar{\beta}_u$  is the density of undetected targets. When the GTSD factor of the fused track is multiple-termed, (5.27) is calculated for each term and the weighted sum becomes  $L_L(\tau_1, \tau_2)$  with the new weights for the fused track. The SPTSD component of the fused track is identical to that of the home track.

#### **CASE 4: Global (or Local) Home/Empty Foreign**

When the foreign track  $\tau_2$  is empty, any predecessor track  $(\tau_1 \cup \tau_2)_{i_1}$  is empty except for  $i_1$ . The track-to-track association likelihood is one and the TSD of the home track becomes the TSD of the fused track.

#### **CASE 5: Empty Home/Local Foreign**

This association with an empty home track is always possible. However, since the foreign track has only azimuth information it cannot benefit the local node.

#### **CASE 6: Empty Home/Global Foreign**

The foreign track is retained by the node. This is the case when the node is being alerted by another node for an incoming target.

#### **CASE 7: Empty Home/Empty Foreign**

Theoretically this fusion involves updating the distribution of undetected targets and the expected number of undetected targets by

$$\beta_{ND}(x) = \prod_{i \in I_R} (\beta_{ND}^i(x))^{n(i)} \quad (5.28)$$

where  $\beta_{ND}(\cdot)$  is the density of the undetected target density based on the fused information and  $\beta_{ND}^i$  is the same density based on the information node  $i$  in the set  $I_R$ . Due to the complexity in representing  $\beta_{ND}$  and  $\beta_{ND}^i$  and the calculation in the above equation, however, it may not be worthwhile to retain and update these densities at each information node. We thus have used a constant  $\beta_u$  which is the density of undetected targets per unit area. This approximation corresponds to assuming a constant target flow into any area under consideration.

### 5.1.3 Hypothesis Management

The hypothesis management procedures used in the information fusion process are almost identical to those used in the local data processing, and include hypothesis pruning, hypothesis combining and clustering.

## 5.2 INFORMATION DISTRIBUTION

One advantage of having a DSN is that communication requirements are less stringent since the nodes only communicate processed results and not the raw sensor data. The communication requirements can be reduced even further if the nodes communicate as needed and not according to a fixed schedule. Some broadcast communication policies were described in [10]. Although the basic principles remain the same, the nature of our algorithms also requires the development of new strategies. In the following we discuss some communication strategies implemented in the communication module.

The information contained in each node is in the form of hypotheses. Each hypothesis is assigned a probability and contains a set of tracks. A track can either be local or global. A *local track* is based only on the measurements of one node and is characterized by the estimates and covariances of the azimuth and

azimuth rate. A *global track* is formed after nodes communicate and is characterized by the estimates and covariances of the position and velocity.

The information distributed by each node consists of the hypotheses and the history of communication. The history of communication, as represented by the predecessor nodes of the information graph, is distributed so that each node can reconstruct the partial information graph. This will then be used to determine the fusible hypotheses as well as the redundant information which needs to be removed when hypothesis evaluation and track updating take place.

When a node finishes its local processing, it examines its set of hypotheses to determine whether any communication should take place. Simulation results have shown that frequent communication, if not carefully controlled, may deteriorate the performance since poor information may be transmitted and used. Thus the quality of information is more important than the amount of information.

### 5.2.1 Communication Criteria

In the current simulation of the DSN, the following criteria are used:

1. *Hypothesis Informativeness*; A node should only send hypotheses which are informative. If all the hypotheses are judged to be equally probable, they will not bring much information to the receiver. One measure of informativeness is the entropy of the hypothesis set. Suppose  $\{P_i, i=1, \dots, N\}$  is the set of probabilities for the  $N$  hypotheses, the hypothesis set will be communicated if the entropy is below a certain threshold, i.e.,

$$-\sum_{i=1}^N P_i \log P_i \leq -K \log \frac{1}{N} \quad (5.29)$$

where  $K$  is a positive constant. Note that if only one hypothesis has a very high probability, the entropy will be small and communication takes place.

2. *Track Informativeness*: For each hypothesis which passes the above test, the tracks are tested for their informativeness. The estimated error

covariance (azimuth and azimuth rate) is examined for each local track whose age exceeds a given threshold. The determinant has to be below a given value before communication takes place. The reason for this criterion is that the local track quality has a significant impact on the quality of the global tracks. From past simulations, we discovered that the accuracy of the initial global track obtained by fusing two local tracks is very sensitive to the accuracy of the azimuth and azimuth rate of the local tracks. A poor global track will make future tracking for each node very difficult.

3. *Geographical Location:* Communication takes place only if the receiver can use the information. The position estimates of the global tracks are generated. If the tracks are predicted to lie within the range of another sensor, then communication takes place.
4. *Time since last communication:* If a hypothesis set has been broadcast recently, it does not have to be communicated again since it will not bring much new information to the receiver. This rule is modified as targets move out of the sensor's coverage or into the range of another sensor. If a target is about to move out of a sensor's coverage, the node may broadcast to *hand over* the track to other nodes. Similarly, a node may choose to *alert* another node if the target is about to move into its coverage.

These strategies have been implemented in the current simulation system. The adaptive nature of the communication means that the information graph cannot be specified a priori but has to be generated by each node from the information received.

### 5.2.2 Communication Pattern and Information Graph

For the overall system, the information graph represents the communication that takes place among the nodes and is thus a useful tool in analyzing the communication pattern. Furthermore, when two nodes communicate, the fusion node has to reconstruct the relevant part of the information graph in order to form and

evaluate the fused hypotheses. The fusion process may be simple or complex depending on the information graph. If the nodes have limited memory, then the process may only be sub-optimal for a complex information graph. In this section, we discuss how communication patterns (and thus strategies) can be evaluated by examining the information graphs.

Certain communication patterns can be analyzed by simply examining the flow of information in the information graph. For instance, consider two alternative communication patterns for a two-node system. In the first pattern node 1 transmits to node 2 periodically at the end of every  $N$  sampling intervals. In the second pattern, node 1 and node 2 transmit alternately to one another with the same duration as the first case (see Figure 5-2 for the information graphs). In both patterns, the communication resources utilized are approximately the same and at the end, node 2 eventually has the same information. However, in pattern one, node 1 never receives information from node 2, which will obviously lead to a poorer performance than that in pattern 2. Therefore, pattern 2 is a better communication pattern unless node 1 does not need node 2's help, e.g., the targets are flying from node 1 to node 2.

A good communication pattern can also simplify the fusion process and improve its efficiency and optimality. For instance, consider a three-node system with cyclic communication. The normal three-node cyclic communication pattern leads to the first information graph in Figure 5-3. Suppose that due to limited communication resources, the three nodes do not transmit at the same time. Instead at any given time, only one node can transmit and there is at least one detection/observation time between two communications. Under these circumstances, one will find that there are essentially only two distinct communication patterns, namely, the *semi-cyclic-1* and *semi-cyclic-2* shown in Figure 5-3. Although both cases require the same communication resources, the fusion equations for the two cases derived from tracing back the information graph are quite different. In case 1, we need only trace a few steps to find the information graph and the fusion equations are optimal if each node has a memory of more than four sampling intervals. However, in case two, identifying the common information requires tracing back the information graph to its root. This implies that much more computation and memory are needed to obtain optimal fusion results. One may, of course, impose a limit on the memory to cut down on the amount of tracing, and use the suboptimal fusion equations.

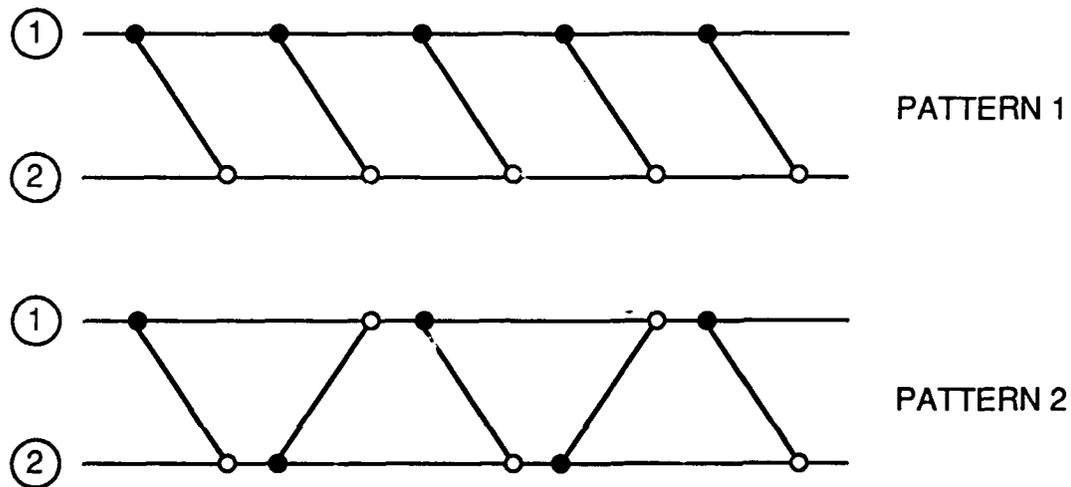


Figure 5-2: Comparison of Two Simple Communication Patterns

From the above examples, we recognize some general principles on the design of communication patterns. In general, the communication pattern should follow the natural information flow in the system as much as possible. To maximize the information flow, information dissemination should be a high priority task of an information sink. For instance, as shown in the semi-cyclic-1 case, whenever a node receives information in the previous scan, it becomes the temporary information sink and should distribute the information to the other nodes as soon as possible. Thus it is important to choose communication strategies that translate into "efficient" information graphs.

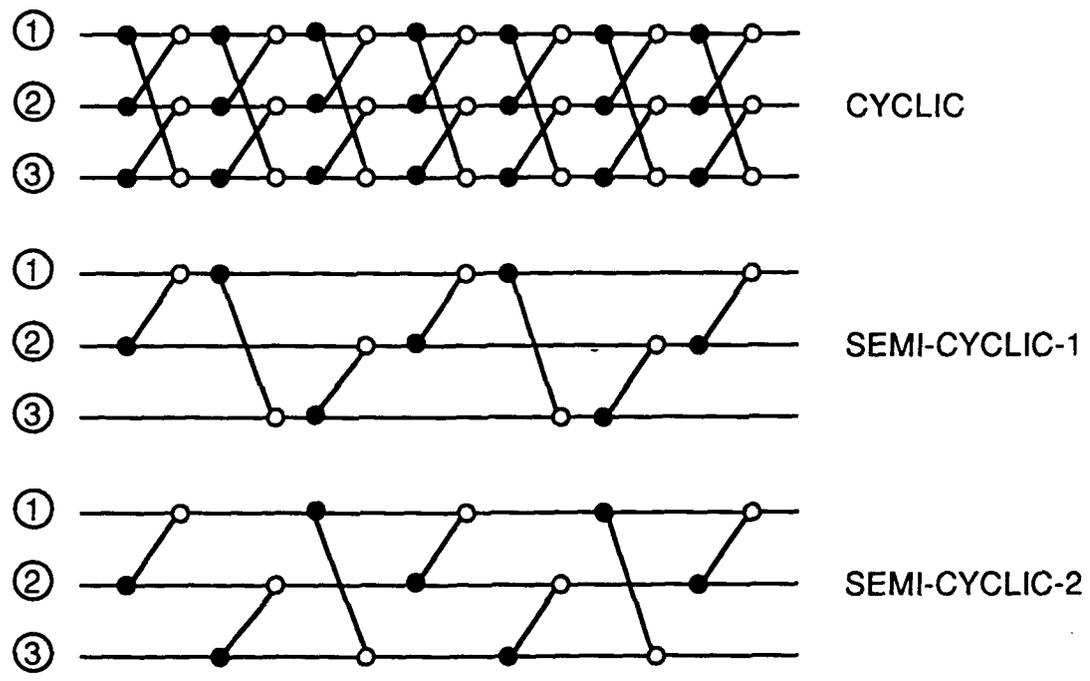


Figure 5-3: Three-Node Cyclic Communication Patterns

## 6. SIMULATION RESULTS

This section presents the simulation results. Section 6.1 describes the simulation environment and Section 6.2 discusses some special features of the algorithms simulated. Sections 6.3 and 6.4 present the simulation results and performance evaluations.

### 6.1 SIMULATION ENVIRONMENT

The simulation was implemented on the Symbolic LISP machine. The software was written in Zetalisp, a dialect of LISP. Figure 6-1 shows the architecture of the simulation environment. The main modules are the data generator, node simulators, simulation controller, communication simulator, and user interface.

#### 6.1.1 Data Generator

We installed the synthetic data generator provided to us by Lincoln Lab. on our Pyramid computer. Data files were transferred to the Symbolics Lisp Machine for display and processing. In order to provide more flexibility in experimentation, a LISP version of the synthetic data generator was implemented on the Lisp Machine. This data generator uses the same models used by Lincoln Lab. The target trajectories can be specified arbitrarily to handle maneuvers if necessary. Each sensor supplies the target azimuths and sound pressures generated using relatively realistic sound propagation models. False alarms and merging measurements are also considered. The measurement vector for each detection thus consists of azimuth, sound pressure and a signal-to-noise ratio. The exact models used were described in Section 3.

#### 6.1.2 User Interface

The user interface module includes different ways of displaying the scenario, the data, and processing results graphically. Scenario displays include sensor locations, coverages and true target trajectories (Figure 6-2). The sensors and targets

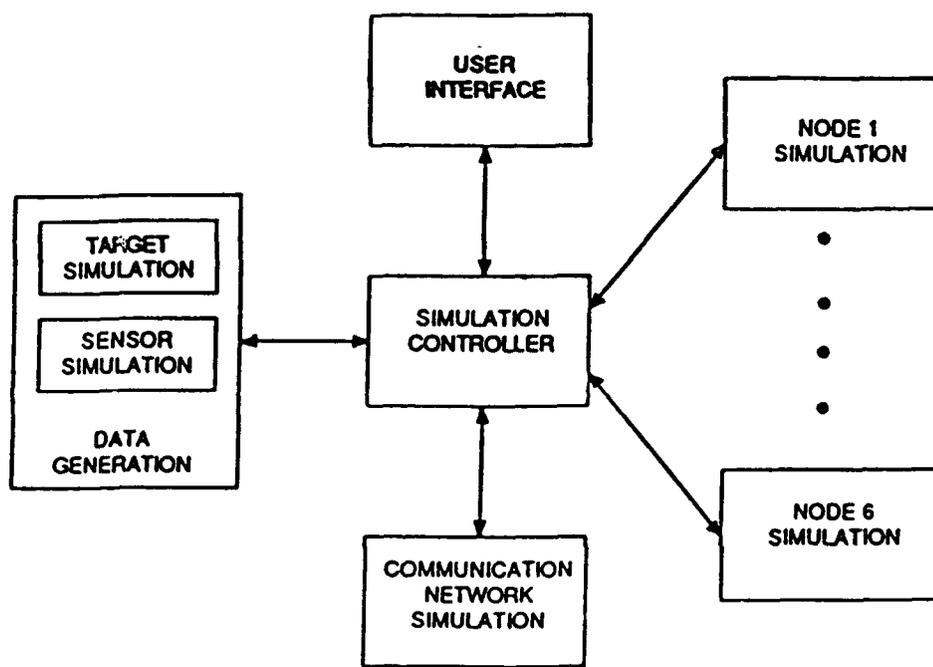


Figure 6-1: Simulation Environment

are mouse-sensitive items. By moving the mouse to the item, it can be described in more detail. For the measurements of each node, the azimuth and sound pressures can be displayed in separate windows. The processing results can also be displayed in the same windows. Figure 6-3 shows the measured azimuths and sound pressures, the estimated azimuths, azimuth rates and sound pressures of the track for three different sensors as compared with the true values. With a color monitor, the values for different nodes are displayed in different colors. A global track display facility is also provided. As shown in Figure 6-4, with this facility we can display the estimated trajectory of the track along with the three-sigma uncertainty covariance ellipses. The true trajectory can also be displayed for comparison. The user can manipulate the windows in various ways, including creation, deletion, freezing, moving, zoom-in/zoom-out, re-scaling, refreshing, and re-shaping. A mouse-sensitive menu has been implemented for selecting the various options.

The key feature of our algorithm is the multiple-hypothesis approach where each hypothesis corresponds to a different explanation of the data. We have implemented mouse-sensitive hypothesis trees and track trees to facilitate the display of the hypotheses. By mousing on a hypothesis (a node on the hypothesis tree), its content and history can be displayed graphically or described in text. Figure 6-5 shows an example where both track trees and hypothesis trees are displayed. The contents of a hypothesis (with one target) are displayed in detail in the three right-hand windows. A textual description of track 166-2 is given in the left-bottom window.

The information flow graph display which illustrates the communication pattern between multiple nodes was also implemented. The information flow graph window was introduced to display the flow of information among the multiple nodes. Since the communication pattern is unknown a priori, this graph allows the pattern to be examined much easier and therefore speeds up the analysis and debugging process. As shown in Figure 6-6, in a three node system, node 1 and node 2 communicate at scan 300 and 314 while node 2 and node 3 communicate at scan 304 and 310. A zoom-out window that gives a better view of the overall communication history is shown in the bottom of the figure.

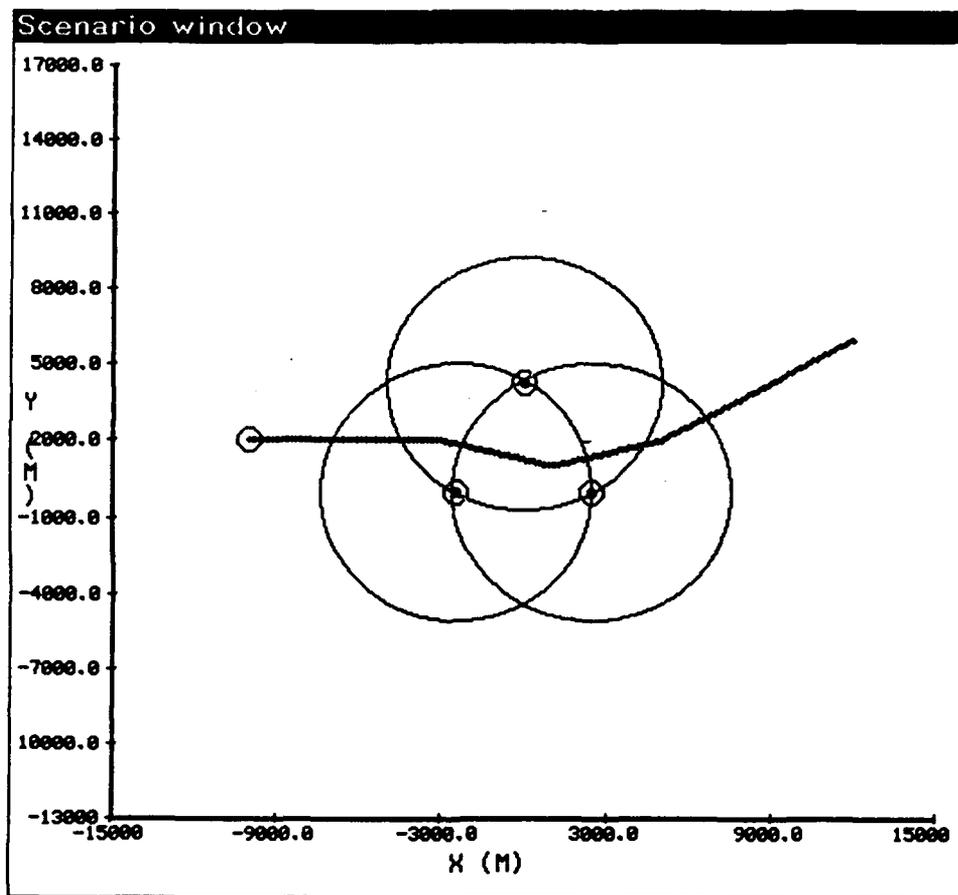


Figure 6-2: Scenario Window

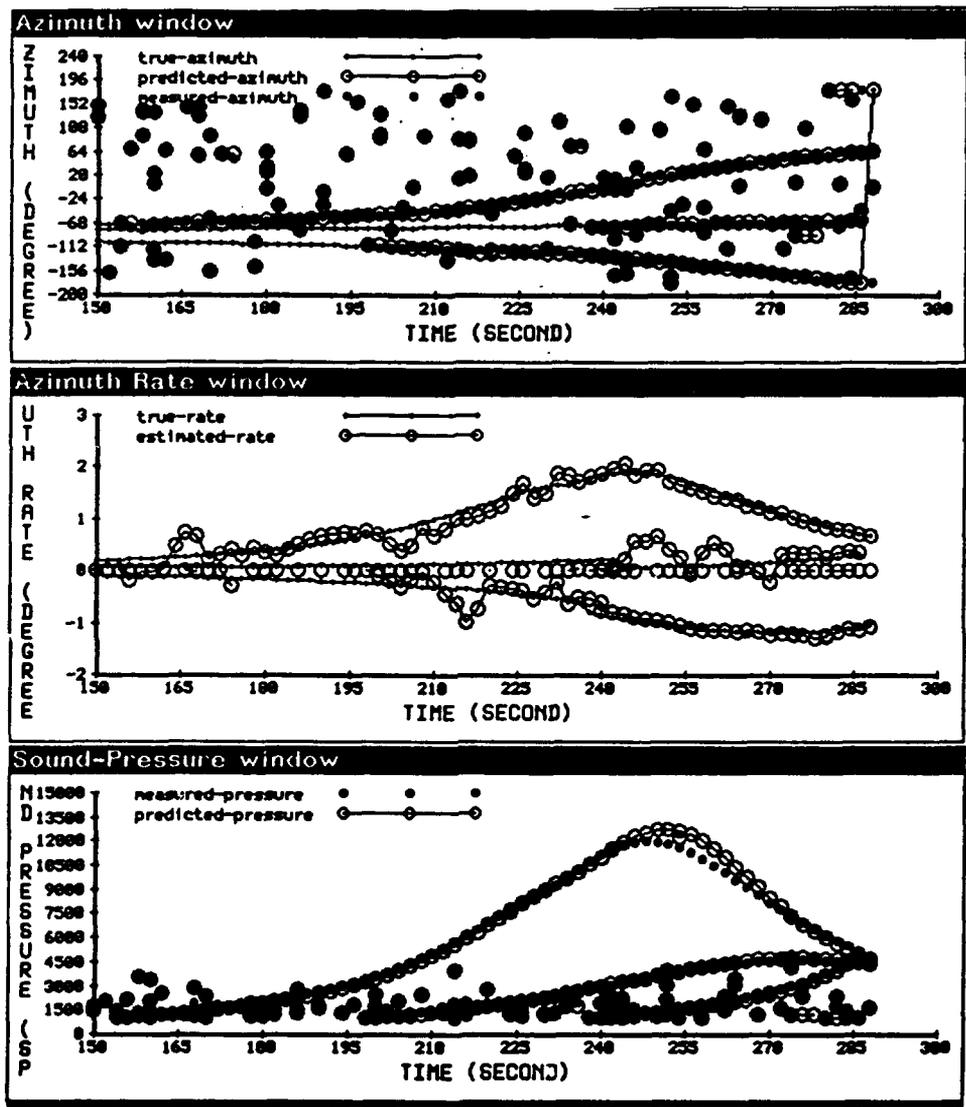


Figure 6-3: Local Track Windows

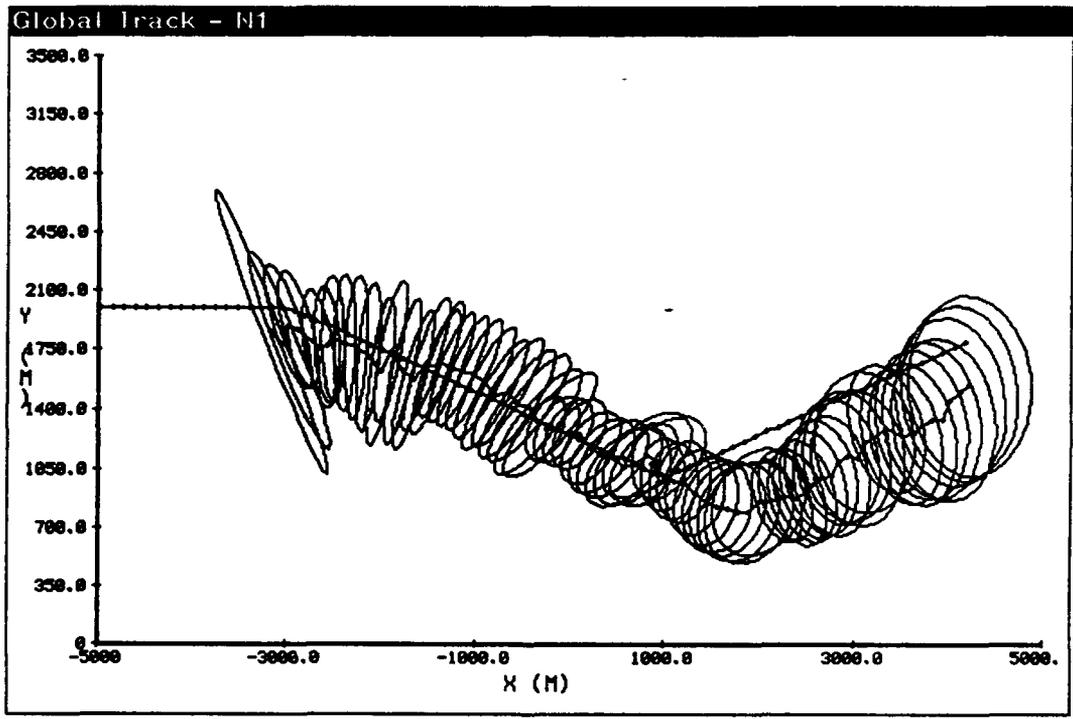


Figure 6-4: Global Track Window

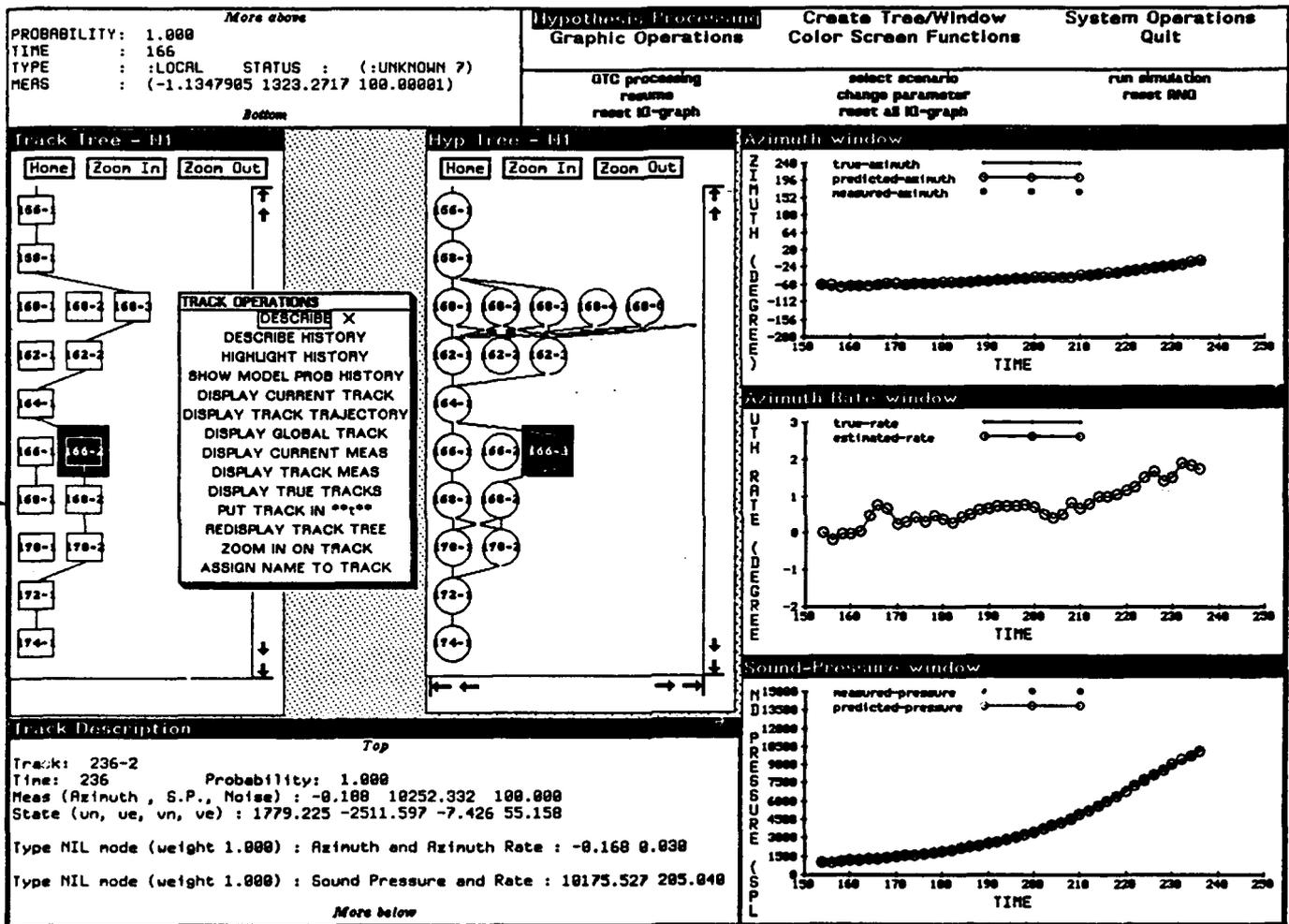


Figure 6-5: Hypothesis and Track Trees

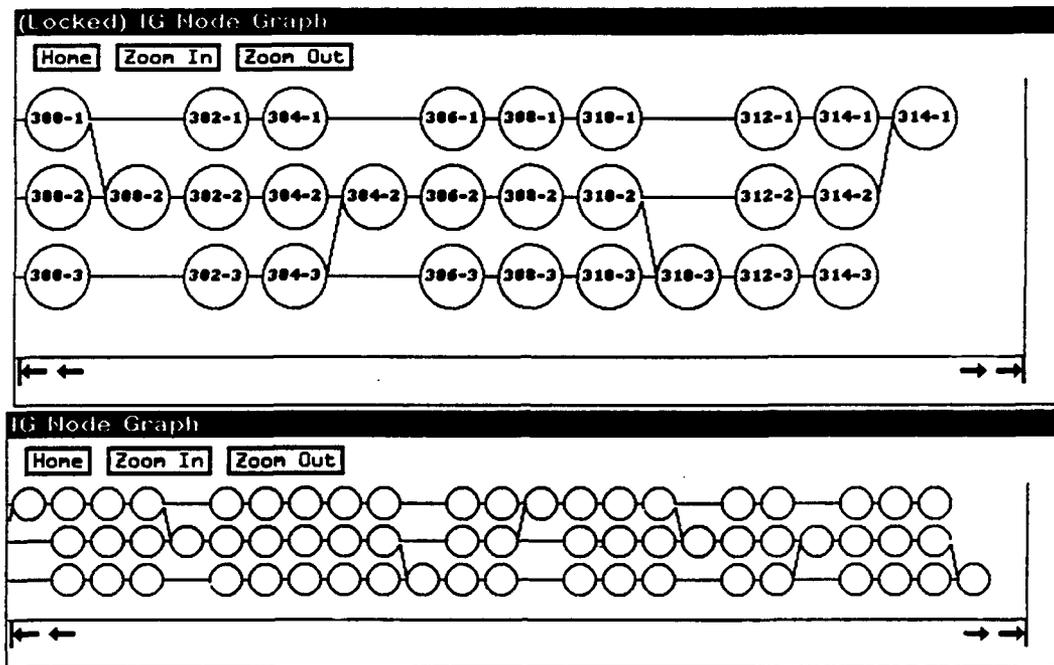


Figure 6-6: Information Flow Graph

### 6.1.3 Simulation Controller

An agenda queue based simulation controller was implemented. A simple simulation control mechanism was designed to be used in the simulation. This mechanism was designed to minimize the overhead while providing enough flexibility to satisfy the needs of the simulation.

At any point of the simulation, we maintain an agenda list. Each agenda contains the following:

1. time of execution
2. priority (coded by a positive integer)
3. action to be taken (coded by a simple function)

At the completion of each action, an agenda control function looks up the agenda list and determines which agenda should be executed next according to the time

and the priority attached to each agenda. Then the top agenda item is executed by transferring the control and executing the action attached to the agenda. Figure 6-7 displays the control flow.

With each execution, an appropriate module performs the necessary simulation action. For example, when a sensor is activated, a set of acoustic measurements is generated and fed into the local information processor. The results are then examined by a communication control function which decides if the particular node should initiate information distribution to other nodes or not. Any function invoked by the agenda execution can access the agenda queue. In the above example, if the communication controller decides to initiate information distribution, it creates an agenda in which the nodes receiving the information fuse the incoming information with their own information states.

Some operations in the simulation have fairly high frequencies. For example, an acoustic sensor sends measurements each two seconds. To handle such repeated operations efficiently, a special kind of agenda, called *scheduled agenda*, was constructed. This kind of agenda item stays in the agenda queue indefinitely until the termination conditions are satisfied.

#### 6.1.4 Communication Controller

A data-driven communication controller was implemented. The basic idea is to select the communication strategy according to the information content. Several rules have been used to make the communication decision. For example, in order to have a better initial global track, local tracks will be communicated between nodes only when they pass some quality checking thresholds (e.g., determinant of the estimated state error covariance). Another example is to check the actual location of the global tracks; only when the global tracks are inside the fields-of-view of both nodes will they be exchanged (see Section 5.2 for more detailed discussion).

There are four objectives of communication: initiation, alerting, information refinement, and hand-over.

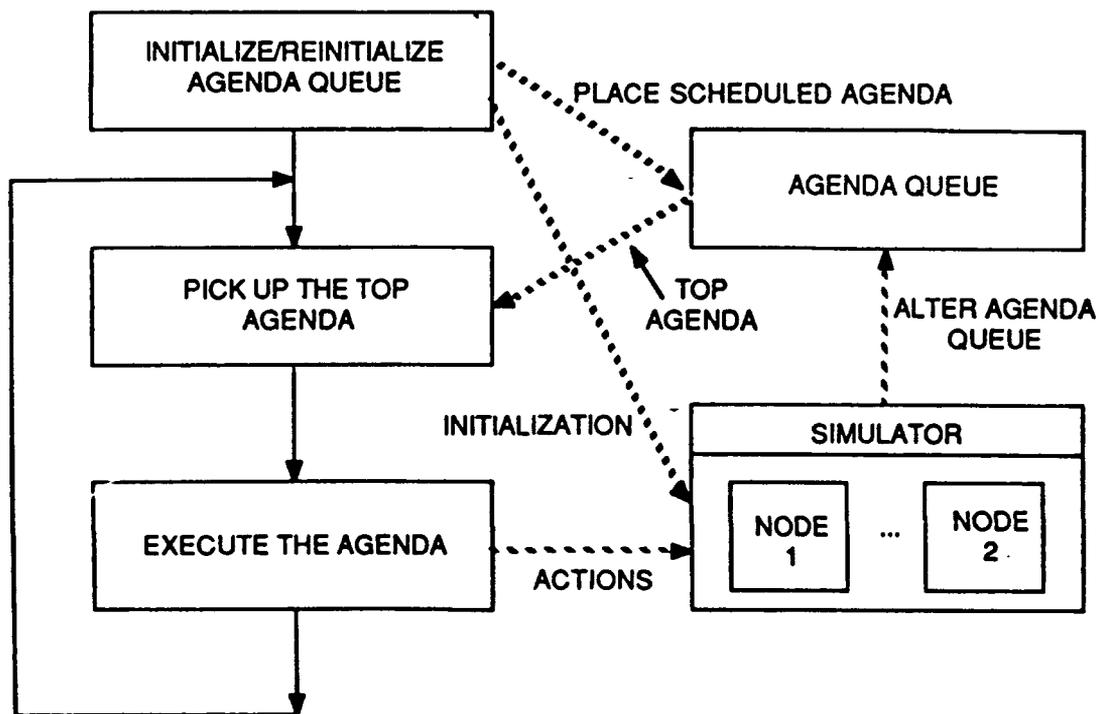


Figure 6-7: Agenda Queue Based Simulation Control

- **Initiation:** Since each node only has azimuth measurements, two nodes are needed to locate a target. Two nodes communicate the local tracks and initiate the global tracks by solving a quadratic equation based on the azimuth and azimuth rate estimates of the local tracks. Once the global tracks are set up, actual target dynamics can be used to extrapolate the target trajectories and future measurements can be used to further refine the estimates.
- **Alerting:** One node sends to the neighboring nodes the global tracks which are moving into their fields-of-view. With the alerting information, the receiving nodes are able to pick up the incoming target estimates faster and more accurately.
- **Information Refinement:** Two nodes exchange and refine their estimates of the global tracks which are inside their common fields-of-view. By looking for confirming information, the false hypotheses and tracks can also be identified and deleted easier.
- **Hand Over:** When a track moves out completely from one node's field-of-view, that track is said to be handed over to the next node and will no longer be kept by the first node. In this manner, each local node only needs to concentrate on its territory and coordinate with other nodes to cover the whole scenario.

Since the communication strategy is data-driven, it depends on the scenario and is unknown a priori. This may perform better than a fixed strategy which may not be adaptive.

## **6.2 ALGORITHM FEATURES**

The distributed tracking algorithms used are those described in Sections 2, 4, and 5. The following, however, are special features that deserve some discussion.

### 6.2.1 Use of Sound Pressure Track

The sound pressure estimate was included in the local track description for the following reasons:

1. To predict detection probability - Since the detection probability depends mainly on the sound pressure, a sound pressure estimate will help in predicting detection probability and providing better data association likelihoods.
2. To identify false measurements - Based on the continuity argument, if a target was detected (heard) at a particular time, it should be detected during a period of time around that moment. A suddenly detected measurement with high sound pressure is likely to be a false alarm.
3. To better understand the track status - A track with gradually increasing sound pressure estimates indicates that the target is approaching. On the other hand, a track with decreasing sound pressure estimates means the target is leaving. Eventually, when a target moves out of the sensor detection range, we can predict the situation and hand the track over to the neighboring nodes with the help of the sound pressure estimates.
4. To help in evaluating merged measurement hypotheses - When two measurements are merged, the corresponding sound pressures are also superposed. A sudden jump of the sound pressure from two lower ones merging into one or a sharp decrease of the sound pressure from a higher one splitting into two lower ones indicate the merging or splitting of measurements. Together with the azimuth information, a good sound pressure estimate may significantly improve the prediction and evaluation of the merged measurement hypotheses.

### 6.2.2 Global Track Initiation

Due to the problem characteristic, it is found that some parameters in the tracking algorithm, especially the process noise, have significant impact on the tracking performance. For example, in the scenario given in Figure 6-8, a two-sensor DSN is tracking a single target with constant speed of 0.2 Mach. The local azimuth rate estimates using a single sensor with a process noise variance ( $Q$ ) of  $1.0e-8$  are shown in Figure 6-9. The results with the same data but with a different  $Q$  of  $1.0e-6$  are given in Figure 6-10. Clearly, the process noise of the first case is a closer match to the real model and will result in better performance.

Based on the simulation experience, the accuracies of local azimuth and azimuth rate estimates have a significant impact on global track initiation. For example, one may initiate a global track using either the local tracks obtained in Figures 6-9 or 6-10. In the first case when the two local tracks from Figure 6-9 are fused, the resulting global state estimate is fairly accurate and the true state is well within the 3-sigma covariance ellipse (see Figure 6-11). However, in the second case, when the two local tracks from Figure 6-10 are fused, the results are much poorer. As one can see, the covariance ellipse is much bigger and the estimated position is far away from the true position (Figure 6-12).

### 6.2.3 Multiple Model Approach

As mentioned in Section 4, a *multiple model* approach is useful in tracking maneuvering targets or multiple targets with different speeds. For example, the tracking results and the corresponding model probability history for a target with speed gradually increasing from 0.2 Mach to 0.8 Mach is shown in Figures 6-13 and 6-14. In this example, the process noise intensities corresponding to models 1 to 3 are  $1.0e-8$ ,  $1.0e-7$ , and  $1.0e-6$  respectively. As one can see, while the target is increasing its speed, the dominant model shifts from model 1 to model 2, and finally to model 3.

Similar concepts have been applied to sound pressure tracks. While the target is far away from the sensor, the sound pressure variations are small and small process noise is sufficient for the filtering. When the target moves closer to a sensor, a larger process noise is then needed to model the large variations of the sound pressure.

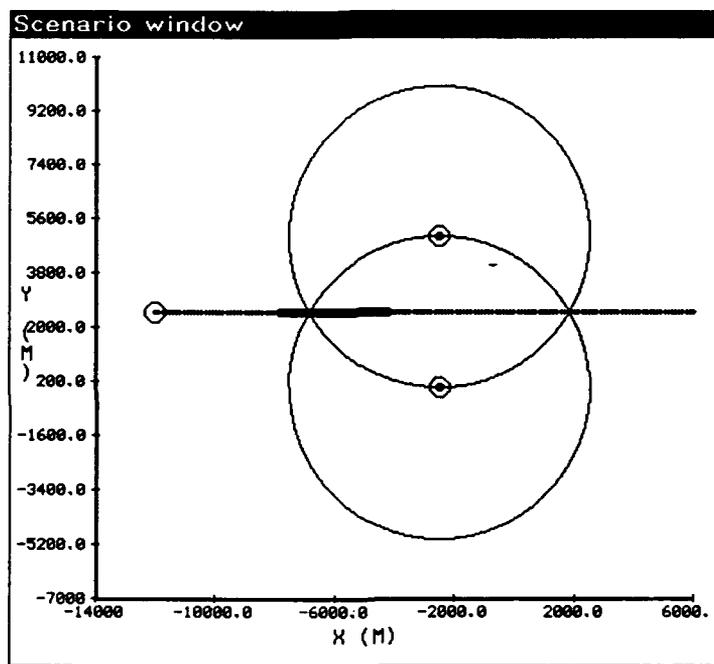


Figure 6-8: Scenario Window

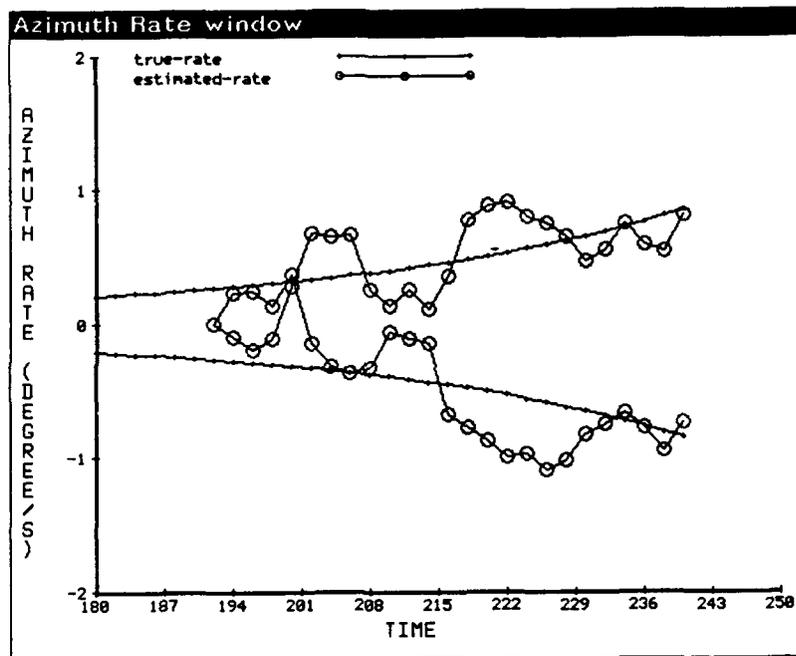


Figure 6-9: Local Tracking Results with  $Q=1.0e-8$

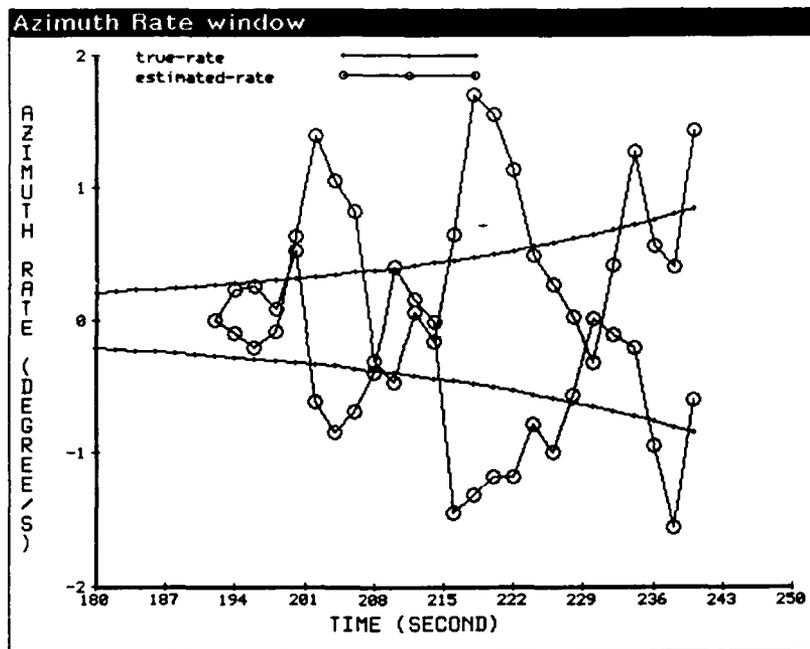


Figure 6-10: Local Tracking Results with  $Q=1.0e-6$

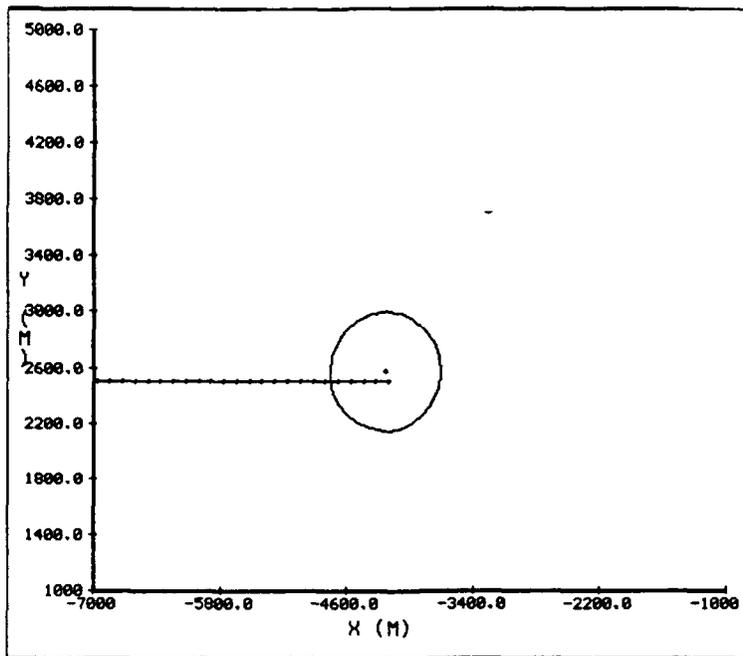


Figure 6-11: Case I : Initial Global Track

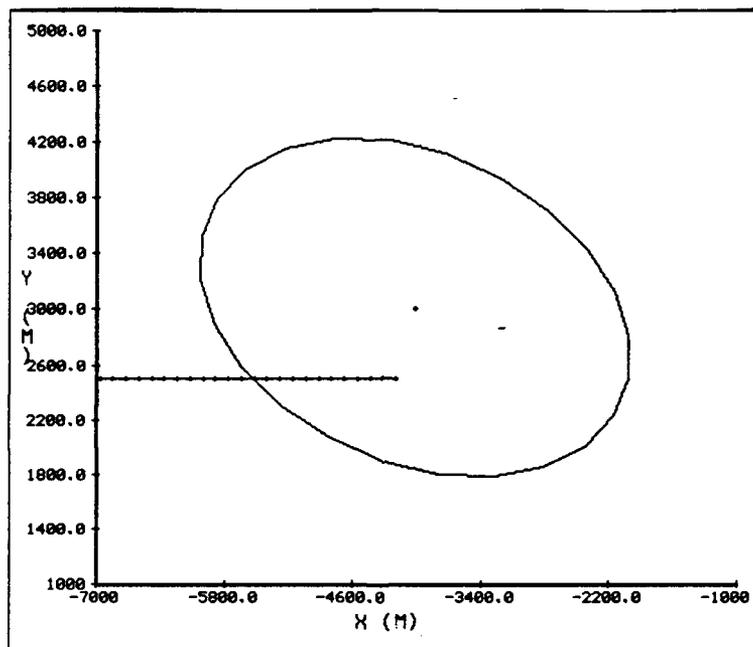


Figure 6-12: Case II : Initial Global Track

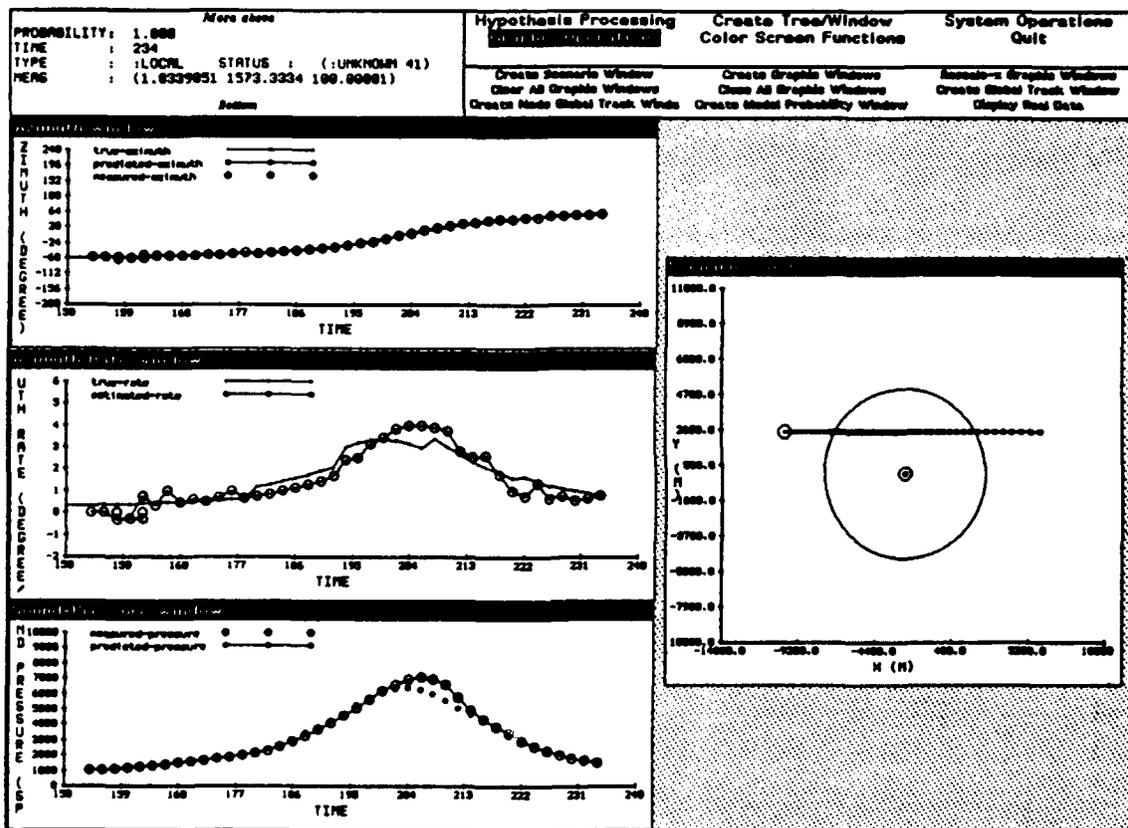


Figure 6-13: Local Tracking Results with Multiple Model Approach

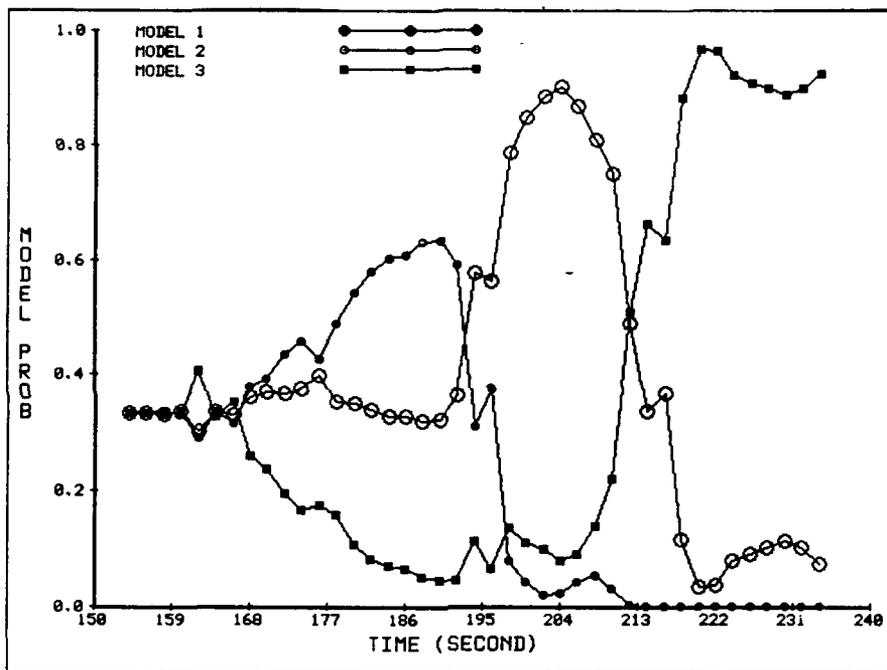


Figure 6-14: Model Probability History

#### **6.2.4 Merged Measurement Models**

The merged measurement problem causes difficulties in both estimating the target state and the number of targets. Since targets gradually resolve while they move closer to the sensor, it is critical to communicate the right information at the right time especially for the first few detecting sensors. For simplicity, only two-way merging was assumed initially in the algorithm. A heuristic rule was added later to handle the possibility of multiple targets merging.

#### **6.2.5 Selection of Communication Strategies**

Simulation results showed that frequent communication, if not suitably controlled, will only deteriorate the performance since poor information may be used. Thus, the quality of information is more important than the amount of information. A smart communication controller is, therefore, very important. Important communication strategies such as track and hypothesis informativeness, information innovation were used in the communication controller (described in Section 5.2).

### **6.3 SIMULATION RESULTS**

Simulations with synthetic data were performed for different scenarios ranging from one to three targets using a DSN with three to seven sensors. Simulations with pre-recorded real data (tracking a single target using two sensors) from Lincoln Lab. was also conducted.

#### **6.3.1 Simulations with Synthetic Data**

There are a large number of possible target scenarios. We identified 20 relatively specific scenarios to focus on. They were selected based upon inputs provided by M.I.T. Lincoln Laboratory.

In the synthetic data generator, the target detection probability depends on the sound pressure measurement, which in turn depends on the target-to-sensor distance and the receiver gain. The false alarm density is assumed to be uniform

in azimuth space and exponentially distributed in sound pressure space. The number of false alarms is assumed to be Poisson distributed. The parameters used in the synthetic data generator are summarized in Table 6-1.

### 6.3.1.1 Single Target Scenarios

Figure 6-15 lists the single target scenarios. Both maneuvering and non-maneuvering situations were considered. Maneuvers include changes of speed and direction. For non-maneuvering targets, we consider both direct and angled approach to the boundary of the distributed sensor network. There is a total of ten scenarios considered in this case.

Table 6-1: Parameters Used in Synthetic Data Generator

$\lambda_{FA}$	$\lambda_{NT}$	$\delta\phi$	$G$	$S_0$	$S_N$
0.5	0.6	0.18 Rad	1.0	106 dB	20 dB

$\lambda_{FA}$ : mean number of false alarms

$\lambda_{NT}$ : mean number of new targets

$\delta\phi$ : sensor resolution

$G$ : sensor gain

$S_0$ : sound pressure at 1 meter from target

$S_N$ : sound pressure of ambient noise

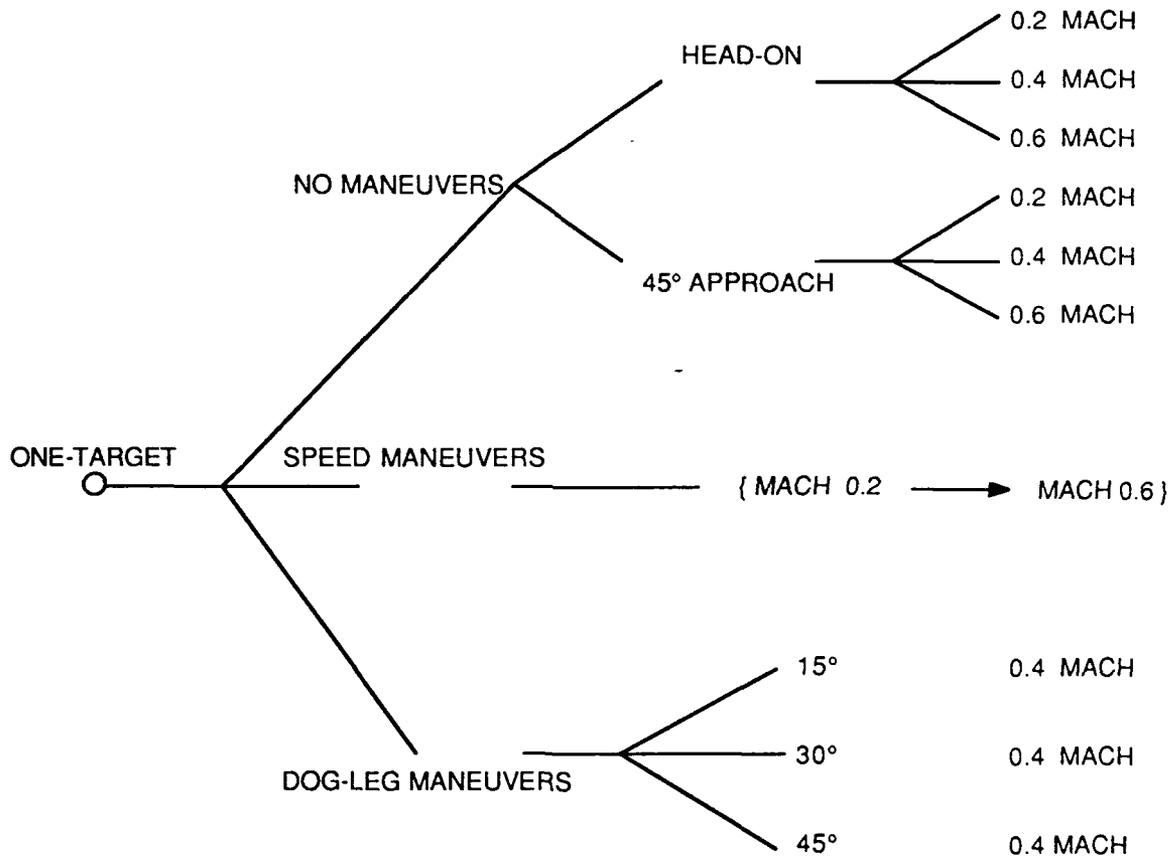


Figure 6-15: Single-Target Scenarios

The first set of scenarios consists of non-maneuvering targets with direct approach to the DSN and three different target speeds. The tracking results are shown in Figures 6-16 to 6-18, where in each figure the trajectories of local azimuth, azimuth rate, and sound pressure tracks are given. The overall scenario and the resulting global track for node 3 (nodes are numbered 1, 2, 3 from left to right) are also shown in each figure. As one can see the algorithm performs well in all three cases. Table 6-2 summarizes the average performance of the last ten scans for each node of each case. In the table five different performance measures are given, namely, the number of tracks, the number of missing targets, the number of false alarms, the positional error, and the velocity error. They were calculated by averaging the results from the last ten scans of each node. The best (highest probability) hypothesis was chosen at each scan. The table indicates that node 3 always has the best performance since it has the benefit of the information from other nodes.

The second set of scenarios deals with similar target configurations but with an angled (45 degree) approach to the DSN. The results are shown in Figures 6-19 to 6-21. In this scenario, the target gets very near to sensor 3 and generates a high sound pressure at the node. Again, in this set of scenarios, the algorithm performs very well. A summary of the performance evaluations is given in Table 6-3.

The third set of scenarios deals with a single maneuvering target. Two types of maneuvers were simulated, namely, speed and direction. In the first case, a target was simulated with a speed changing from 0.2 Mach to 0.6 Mach. Then in the next three cases, a target with a dog-leg maneuver was simulated with a direction change of 15, 30, and 45 degrees. The target speed was maintained at 0.4 Mach. The results are presented in Figures 6-22 to 6-25.

Using the Multiple Model approach described in the previous section, the algorithm tracks the maneuvering target with speed change reasonably well. In the scenarios with dog-leg maneuvers, the cooperation between the sensors is especially important. Because each sensor has a different aspect angle on a target trajectory segment, exchanging information between sensors allows early detection of maneuvers. The performance evaluation summary for the above four cases is given in Table 6-4.

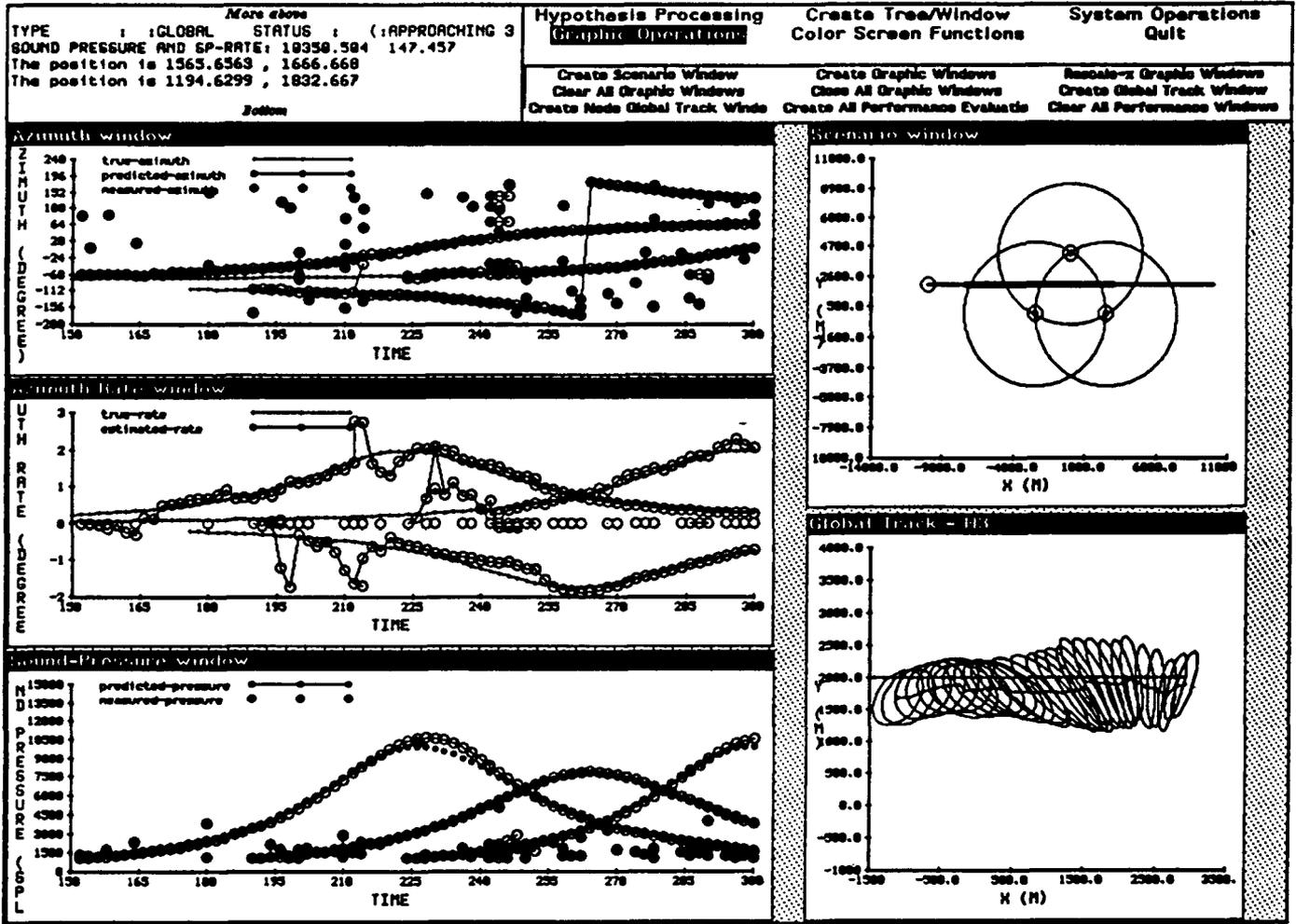


Figure 6-16: Tracking Results with Speed = 0.2 Mach (Scenario 1-1)

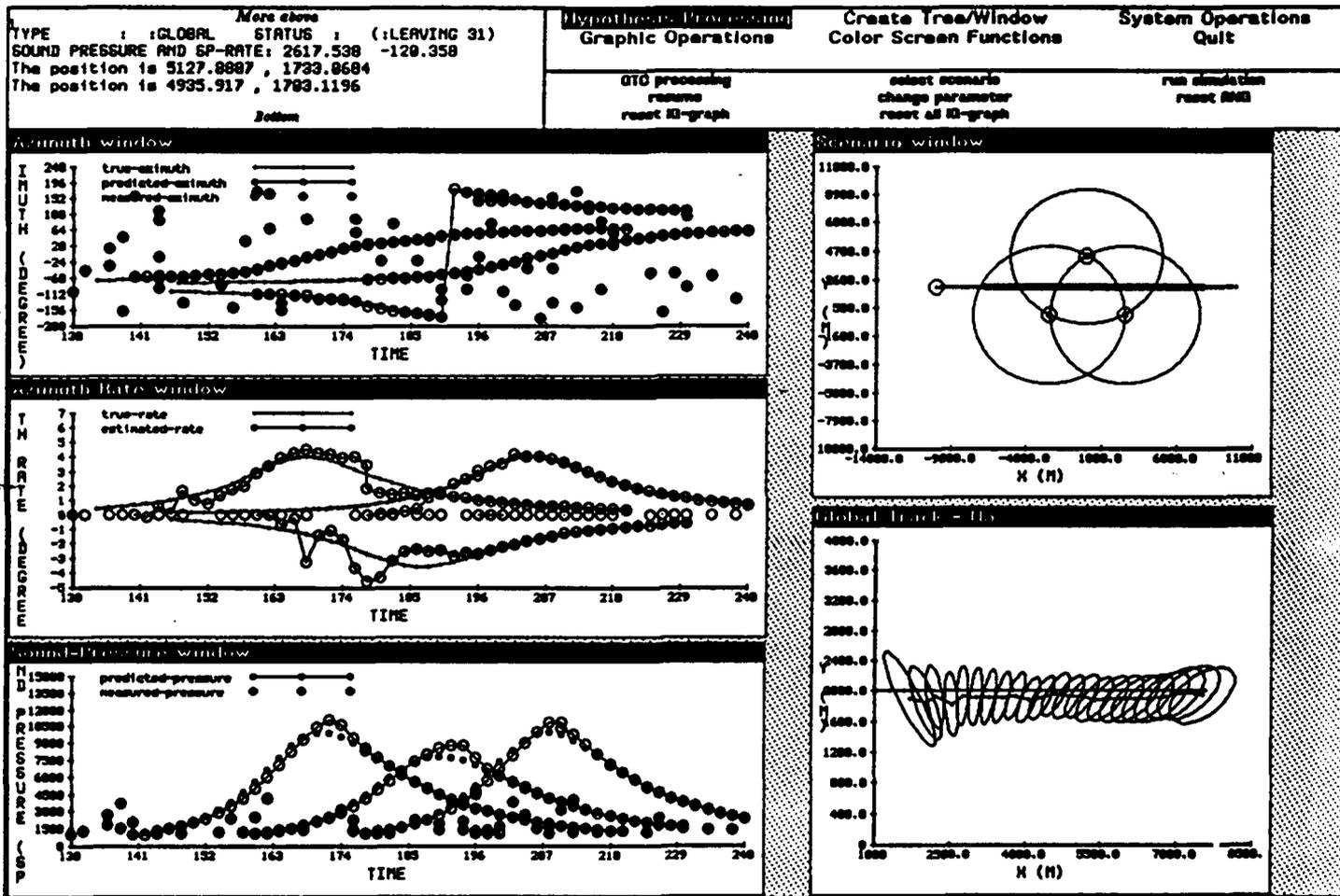


Figure 6-17: Tracking Results with Speed = 0.4 Mach (Scenario 1-2)

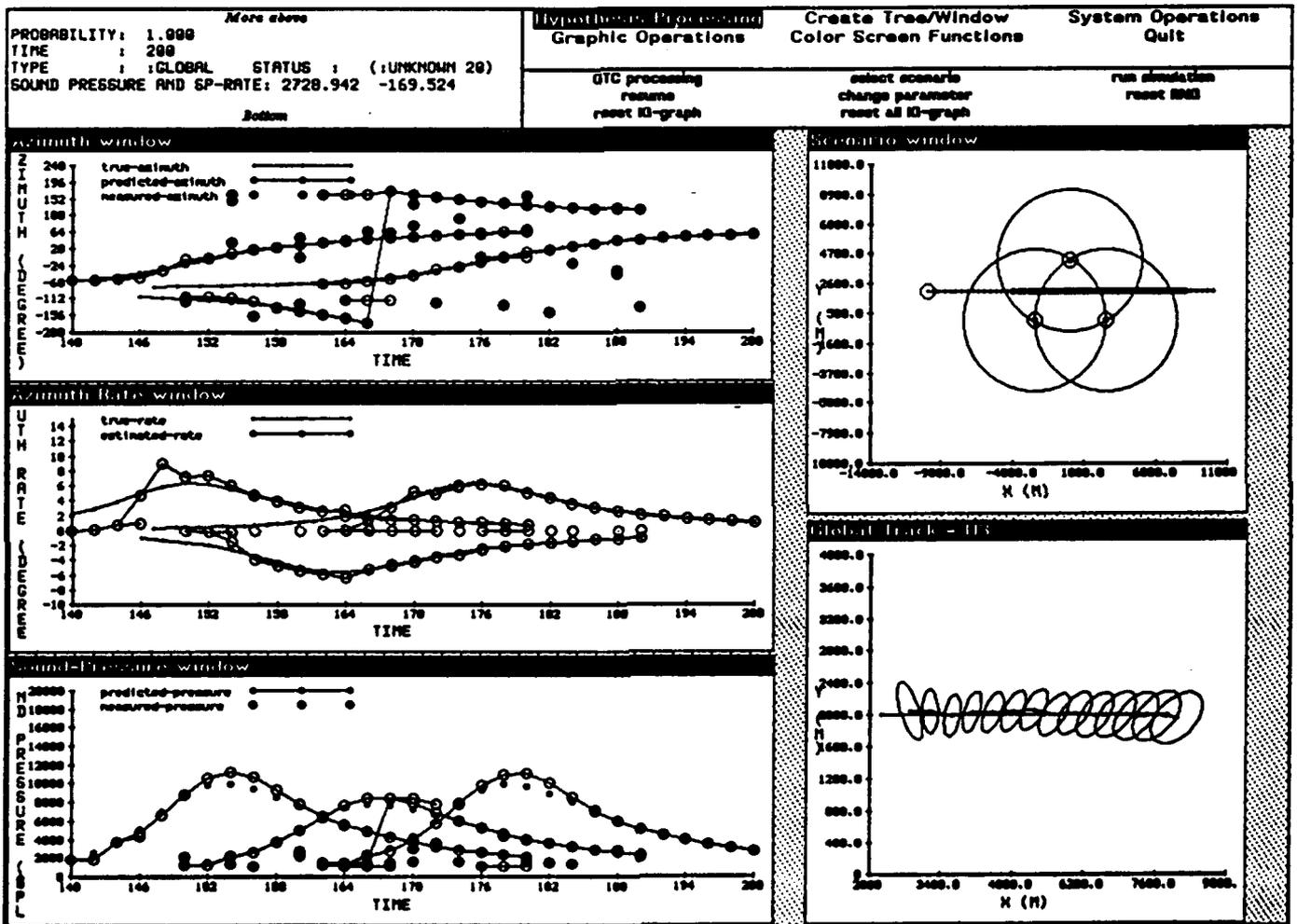


Figure 6-18: Tracking Results with Speed = 0.6 Mach (Scenario 1-3)

Table 6-2: Performance Results - One-Target (I)

Scenario 1-1	Node 1	Node 2	Node 3
# of Tracks	1.0	1.0	1.0
# of Missing Targets	0.0	0.0	0.0
# of False Tracks	0.0	0.0	0.0
Pos. Error(m)	331.84	155.97	135.65
Vel. Error(m/s)	8.73	3.36	1.72
Scenario 1-2	Node 1	Node 2	Node 3
# of Tracks	1.0	1.0	1.0
# of Missing Targets	0.0	0.0	0.0
# of False Tracks	0.0	0.0	0.0
Pos. Error(m)	583.56	259.58	162.25
Vel. Error(m/s)	8.96	7.40	4.41
Scenario 1-3	Node 1	Node 2	Node 3
# of Tracks	1.0	1.0	1.0
# of Missing Targets	0.0	0.0	0.0
# of False Tracks	0.0	0.0	0.0
Pos. Error(m)	344.71	121.43	44.05
Vel. Error(m/s)	23.52	4.07	1.84

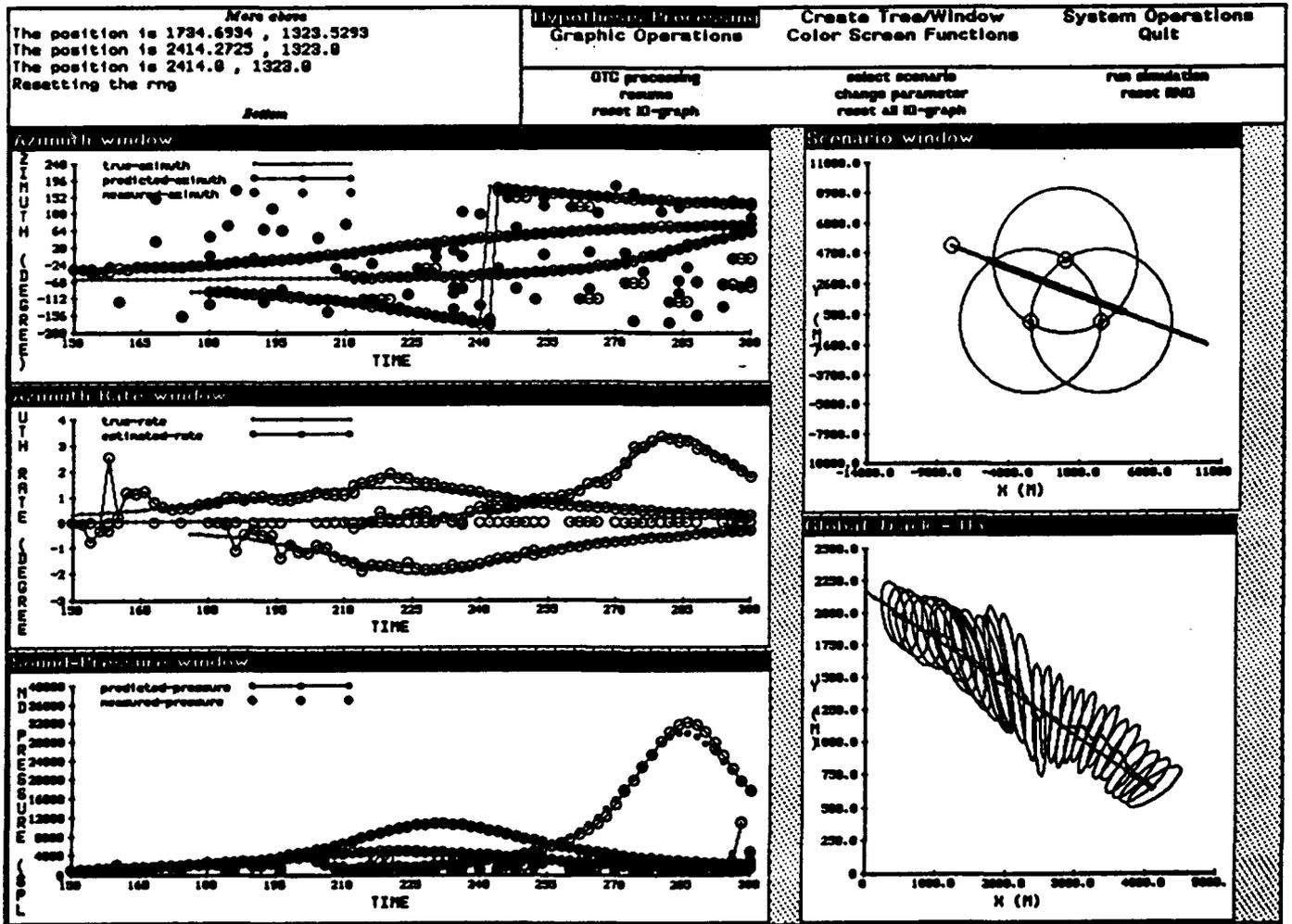


Figure 6-19: Tracking Results with Speed = 0.2 Mach (Scenario 1-4)

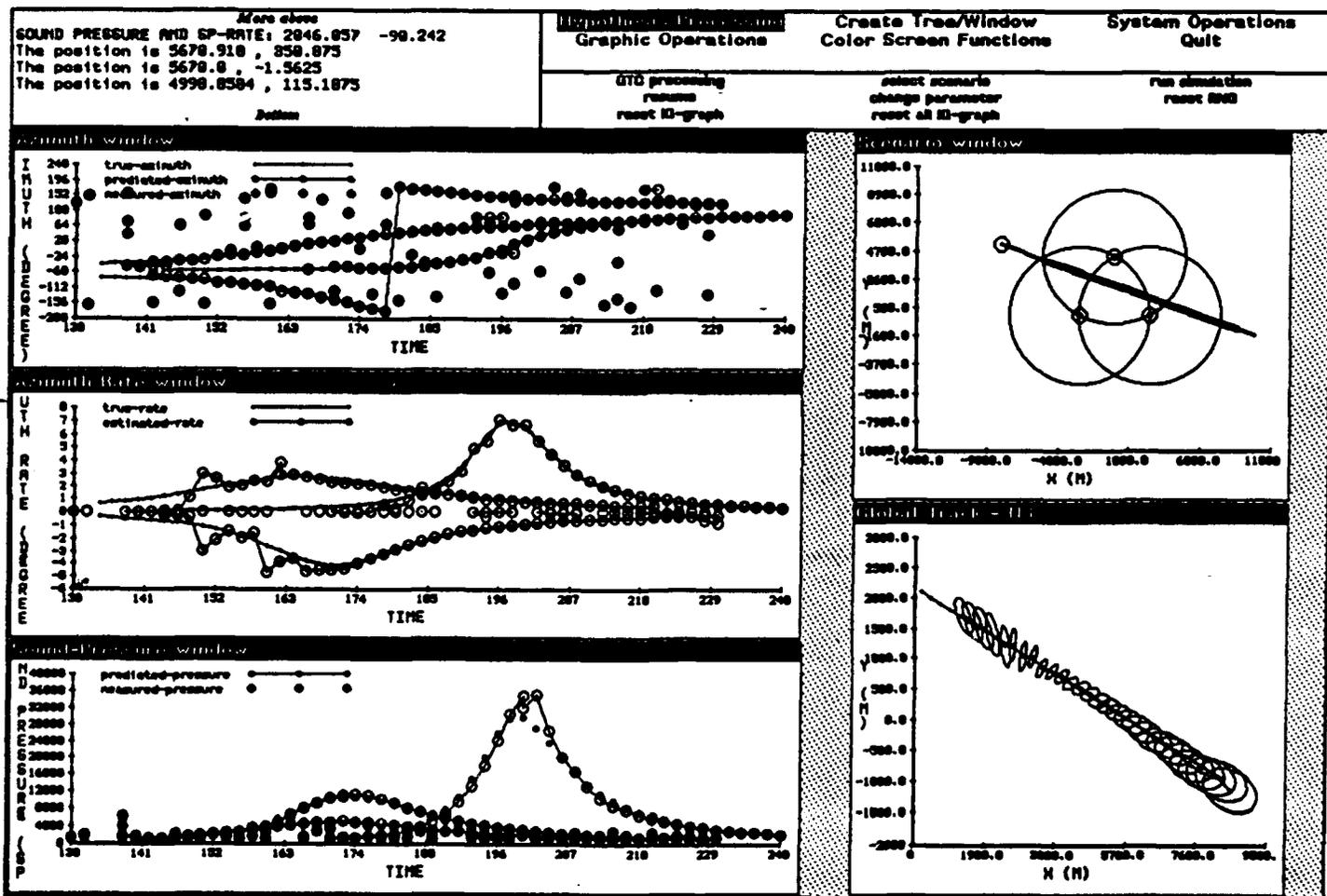


Figure 6-20: Tracking Results with Speed = 0.4 Mach (Scenario 1-5)

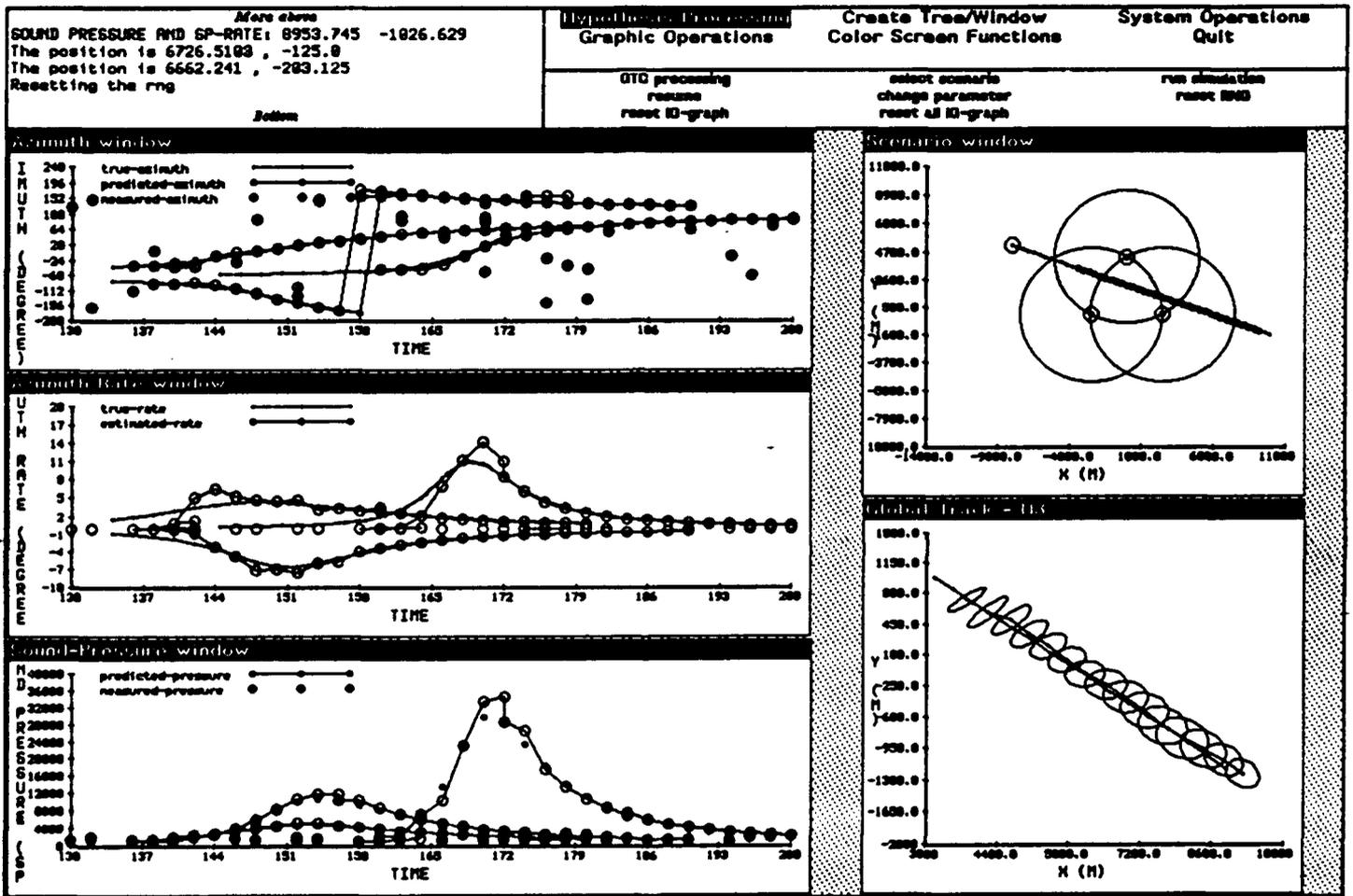


Figure 6-21: Tracking Results with Speed = 0.6 Mach (Scenario 1-6)

Table 6-3: Performance Results - One Target (II)

Scenario 1-4	Node 1	Node 2	Node 3
# of Tracks	1.0	1.0	1.0
# of Missing Targets	0.0	0.0	0.0
# of False Tracks	0.0	0.0	0.0
Pos. Error(m)	343.21	203.01	74.40
Vel. Error(m/s)	6.23	3.62	2.47
Scenario 1-5	Node 1	Node 2	Node 3
# of Tracks	1.0	1.1	1.0
# of Missing Targets	0.0	0.0	0.0
# of False Tracks	0.0	0.1	0.0
Pos. Error(m)	505.57	99.50	57.95
Vel. Error(m/s)	6.07	2.560	2.49
Scenario 1-6	Node 1	Node 2	Node 3
# of Tracks	1.0	1.0	1.0
# of Missing Targets	0.0	0.0	0.0
# of False Tracks	0.0	0.0	0.0
Pos. Error(m)	386.93	374.60	37.41
Vel. Error(m/s)	18.91	11.94	1.54

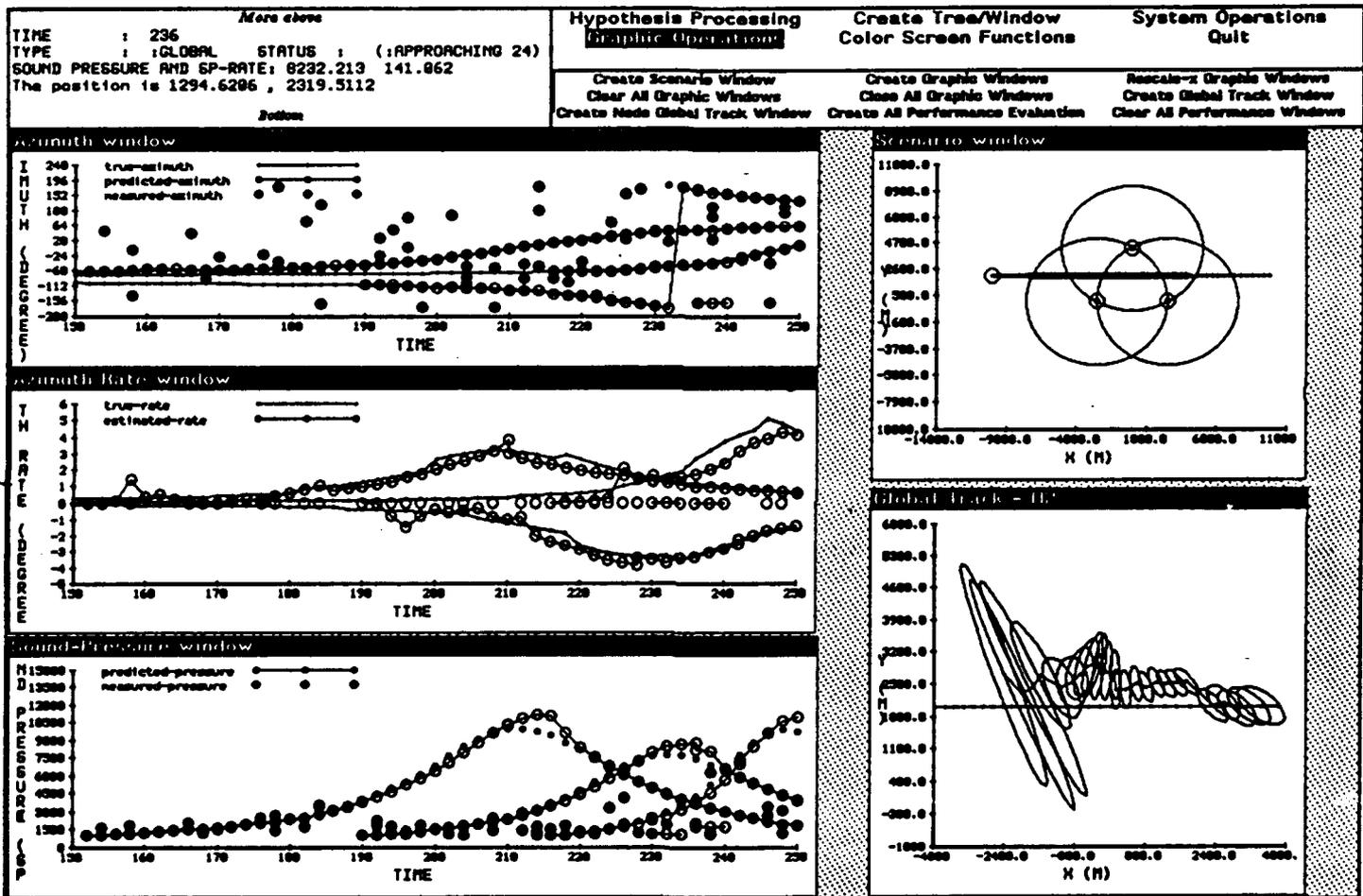


Figure 6-22: Tracking Results with speed from 0.2 to 0.6 Mach (Scenario 1-7)

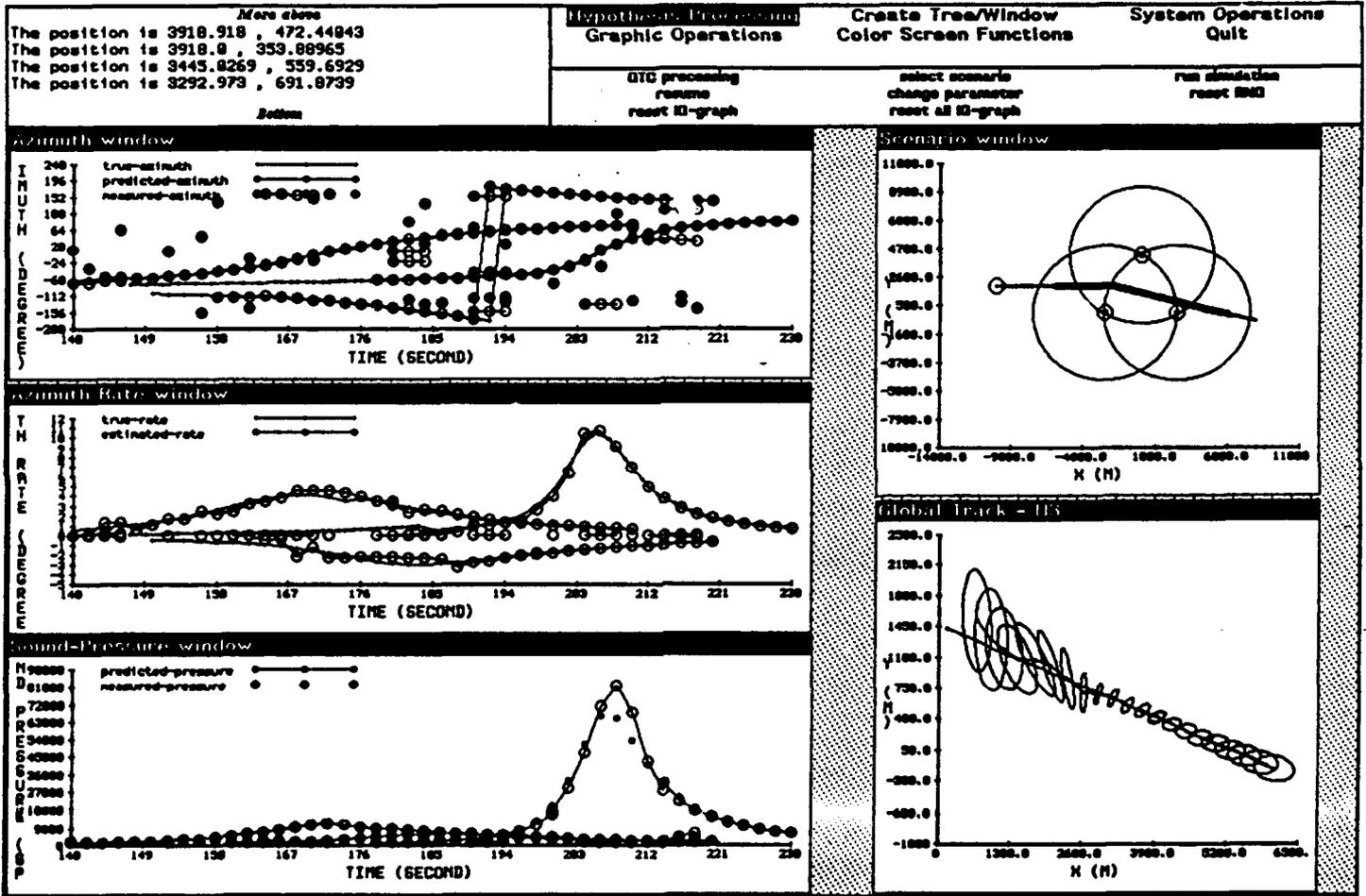


Figure 6-23: Tracking Results with 15 Degree Direction Change (Scenario 1-8)

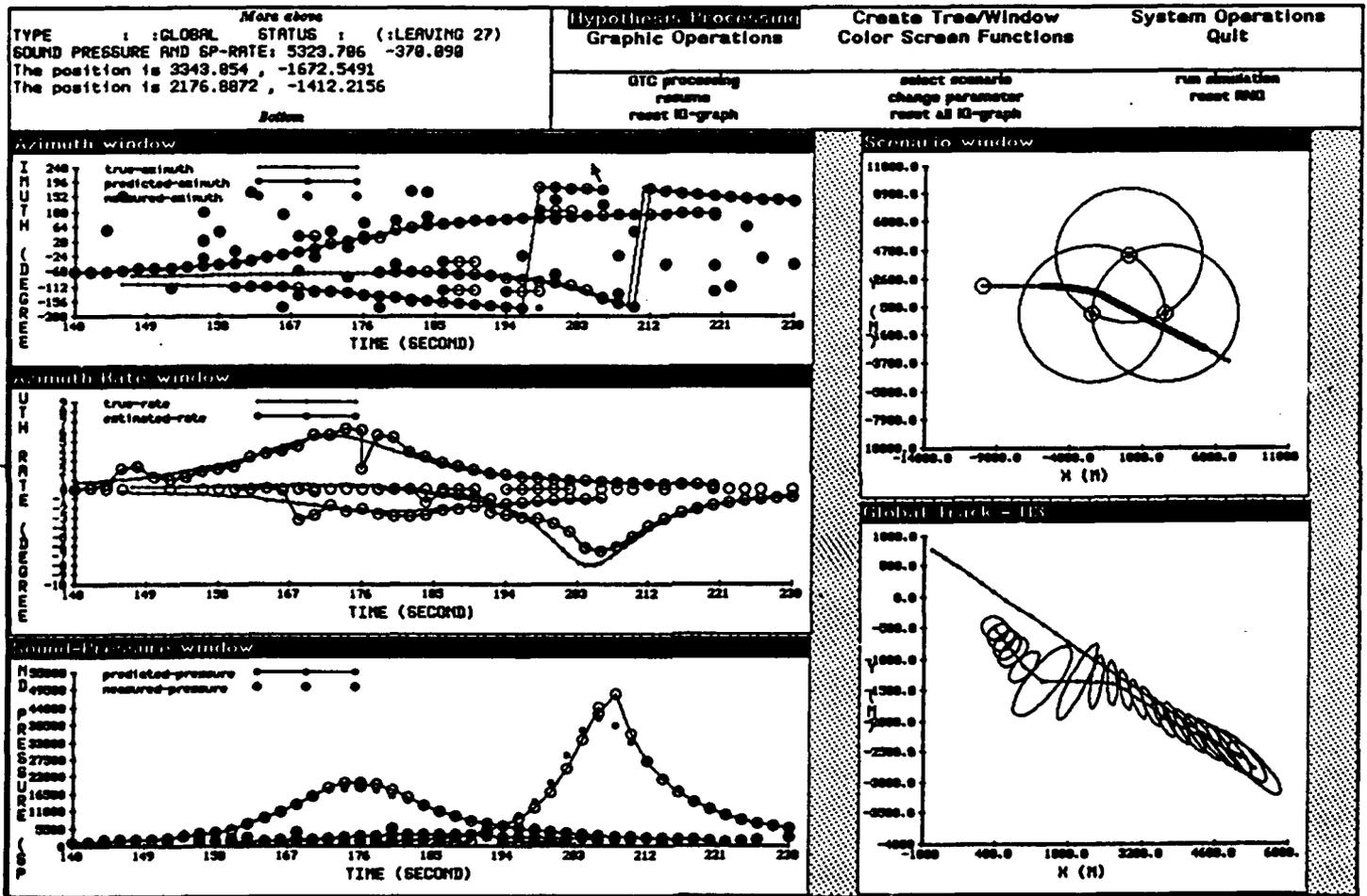


Figure 6-24: Tracking Results with 30 Degree Direction Change (Scenario 1-9)

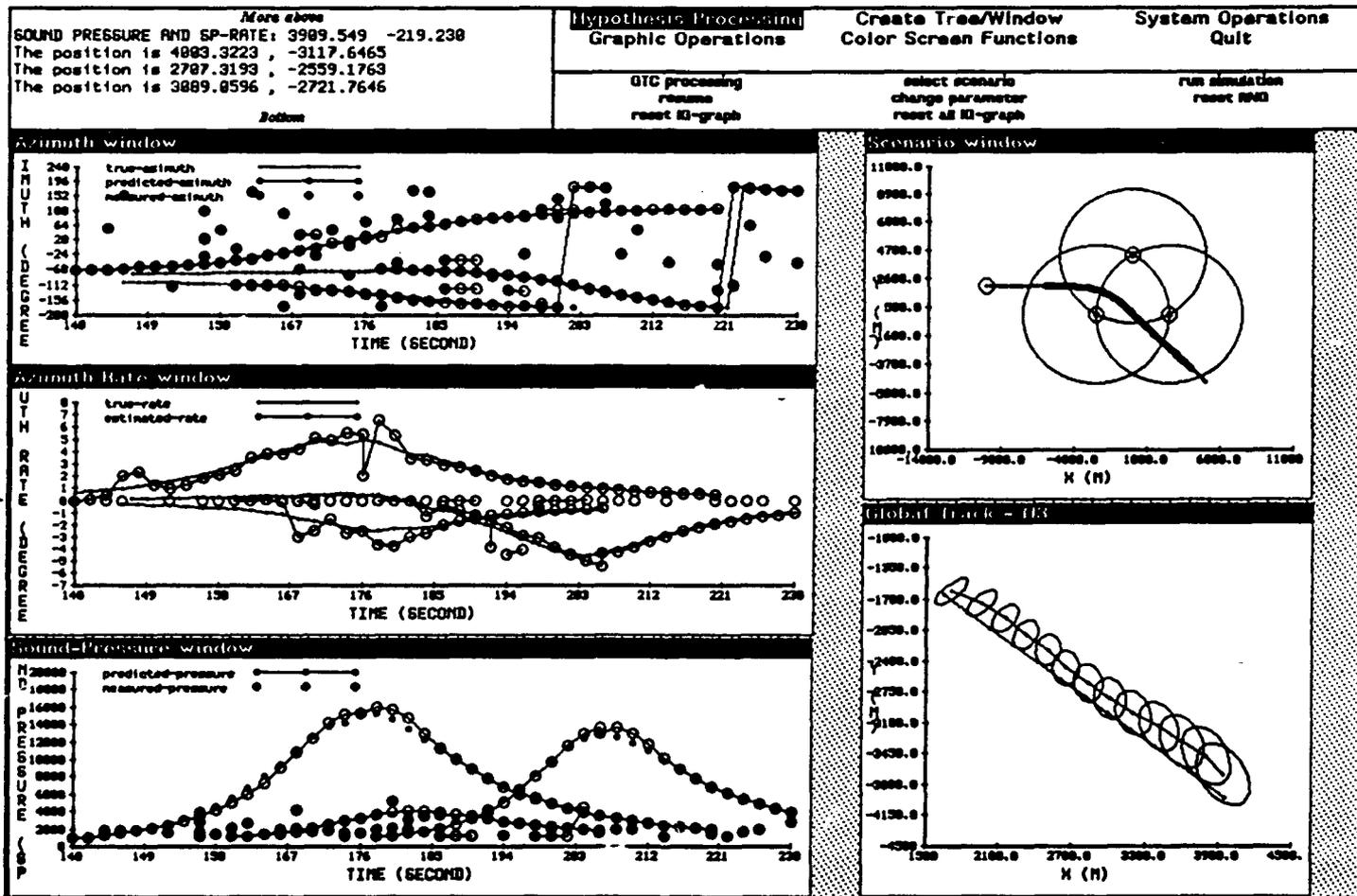


Figure 6-25: Tracking Results with 45 Degree Direction Change (Scenario 1-10)

Table 6-4: Performance Results - One Target (III)

Scenario 1-7	Node 1	Node 2	Node 3
# of Tracks	1.0	1.0	1.0
# of Missing Targets	0.0	0.0	0.0
# of False Tracks	0.0	0.0	0.0
Pos. Error(m)	265.79	522.18	452.13
Vel. Error(m/s)	30.76	37.63	31.33
Scenario 1-8	Node 1	Node 2	Node 3
# of Tracks	1.0	1.0	1.0
# of Missing Targets	0.0	0.0	0.0
# of False Tracks	0.0	0.0	0.0
Pos. Error(m)	850.71	141.91	23.77
Vel. Error(m/s)	23.80	3.59	1.46
Scenario 1-9	Node 1	Node 2	Node 3
# of Tracks	1.0	1.0	1.0
# of Missing Targets	0.0	0.0	0.0
# of False Tracks	0.0	0.0	0.0
Pos. Error(m)	718.21	677.70	206.40
Vel. Error(m/s)	18.39	20.29	6.72
Scenario 1-10	Node 1	Node 2	Node 3
# of Tracks	1.0	1.0	1.0
# of Missing Targets	0.0	0.9	0.0
# of False Tracks	0.0	0.9	0.0
Pos. Error(m)	185.86	1637.97	273.69
Vel. Error(m/s)	6.48	65.59	10.63

### 6.3.1.2 Two Target Scenarios

Figure 6-26 shows a set of two-target scenarios. Three different target configurations were considered: in-line formation, parallel formation, and crossing formation. In each case, both targets have the same speed.

The first two cases deal with the in-line formation, where one target follows the other. The distances between two targets are 1.0 km and 1.5 km and each target has a speed of 0.4 Mach. As seen in Figures 6-27 and 6-28, the two targets were tracked quite well in both cases. Table 6-5 summarizes the performance results for the in-line formation.

In the next few scenarios the parallel formation in which targets follow parallel tracks is considered. In the first two cases, the targets move at a speed of 0.4 Mach and are separated by a distance of 0.5 km and 1.0 km respectively. Due to the poor sensor resolution, when the two targets are separated by less than 0.5 km, only one merged measurement can be detected by the sensors. Therefore, as seen in Figure 6-29, only one track is created and its position estimate falls between the two true target positions. When the two targets are separated by 1.0 km, although resolved measurements can be observed during certain periods for each sensor, they are not strong enough to maintain the two-target hypothesis. As shown in Figure 6-30, the one-target hypothesis still dominates throughout the simulation.

In the last parallel formation scenario, the target speed decreases to 0.2 Mach while the target separation is maintained at 1.0 km. As shown in Figure 6-31, due to more frequent detections of resolved measurements, the two target-hypothesis finally dominates the other hypotheses. The performance results of the above three cases are summarized in Table 6-6.

For crossing targets, two scenarios were considered. In the first case, the targets travel at a speed of 0.4 Mach and in the second case 0.2 Mach. These two cases are interesting not only because of the measurement merging phenomena, but also the configuration which makes association and continual tracking quite difficult. As seen in Figure 6-32, when targets move at a speed of 0.4 Mach, the performance is poor. Although both targets were tracked by node 3 at the end, only one of them was identified by node 3 during at the first half of the simulation. This is due to the severe merging of target measurements that are very

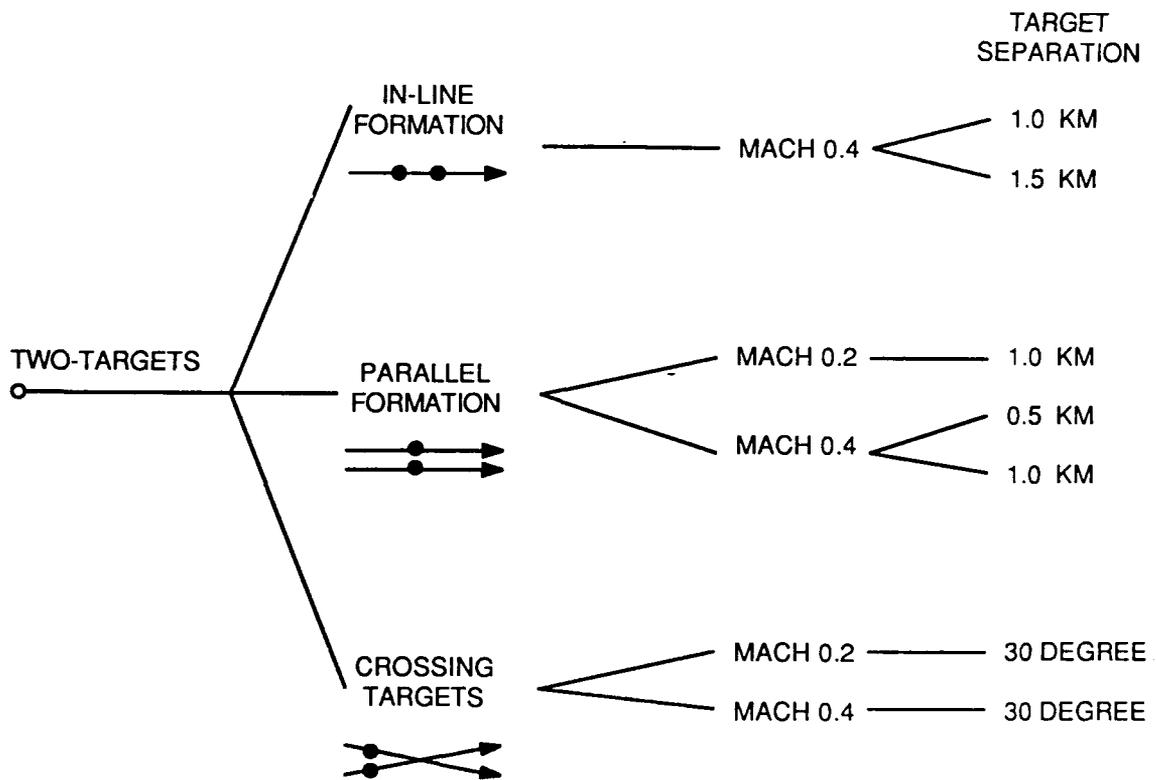


Figure 6-26: Two-Target Scenarios

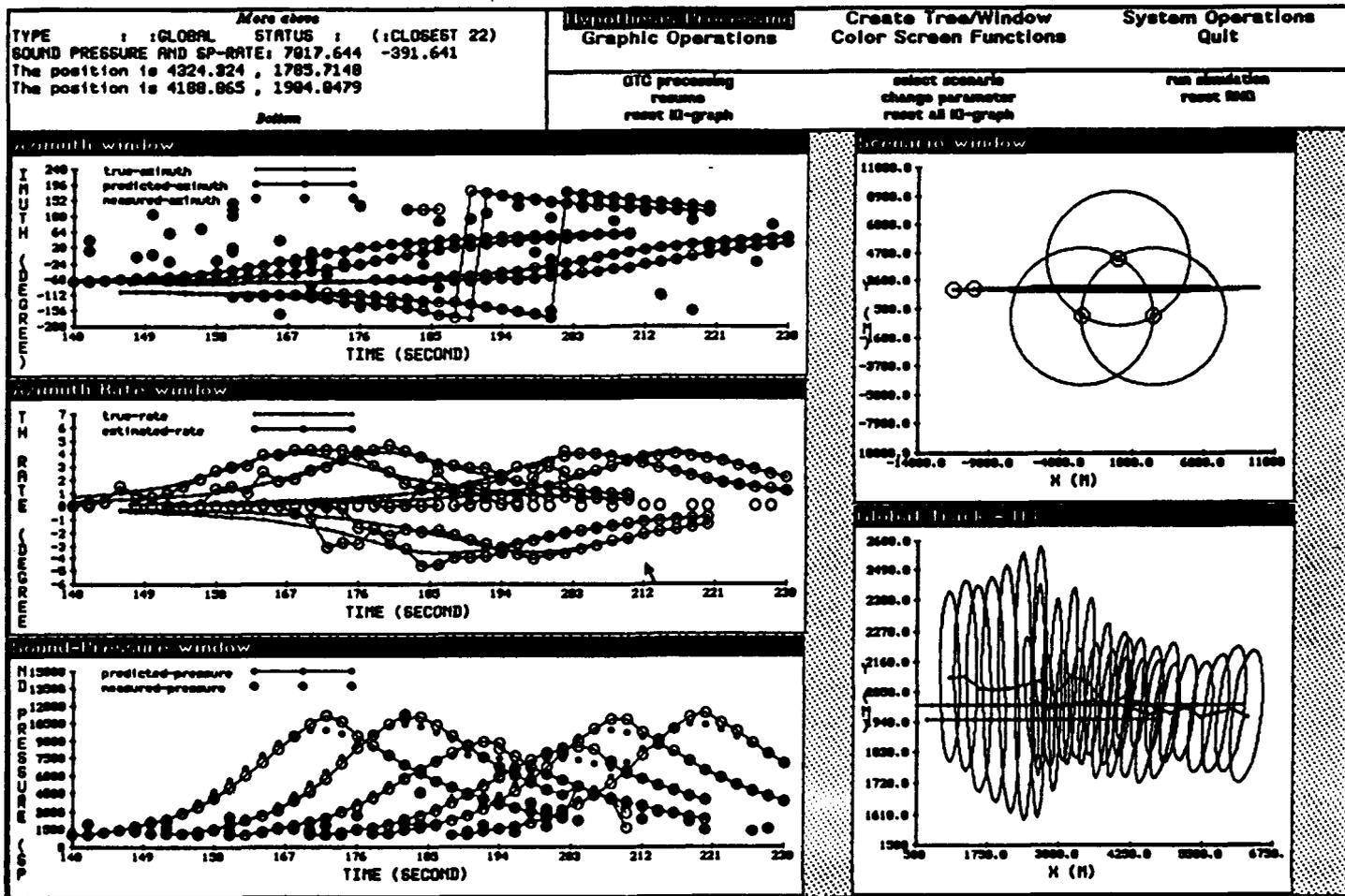


Figure 6-28: Tracking Results of Line-Formation II (Scenario 2-2)

Table 6-5: Performance Results - Two-Target (I)

Scenario 2-1	Node 1	Node 2	Node 3
# of Tracks	2.0	2.0	2.0
# of Missing Targets	0.2	0.0	0.0
# of False Tracks	0.2	0.0	0.0
Pos. Error(m)	1102.47	738.17	236.53
Vel. Error(m/s)	53.35	23.47	6.53
Scenario 2-2	Node 1	Node 2	Node 3
# of Tracks	2.0	2.0	2.0
# of Missing Targets	0.0	0.0	0.0
# of False Tracks	0.0	0.0	0.0
Pos. Error(m)	504.11	229.62	64.68
Vel. Error(m/s)	10.62	3.88	2.32

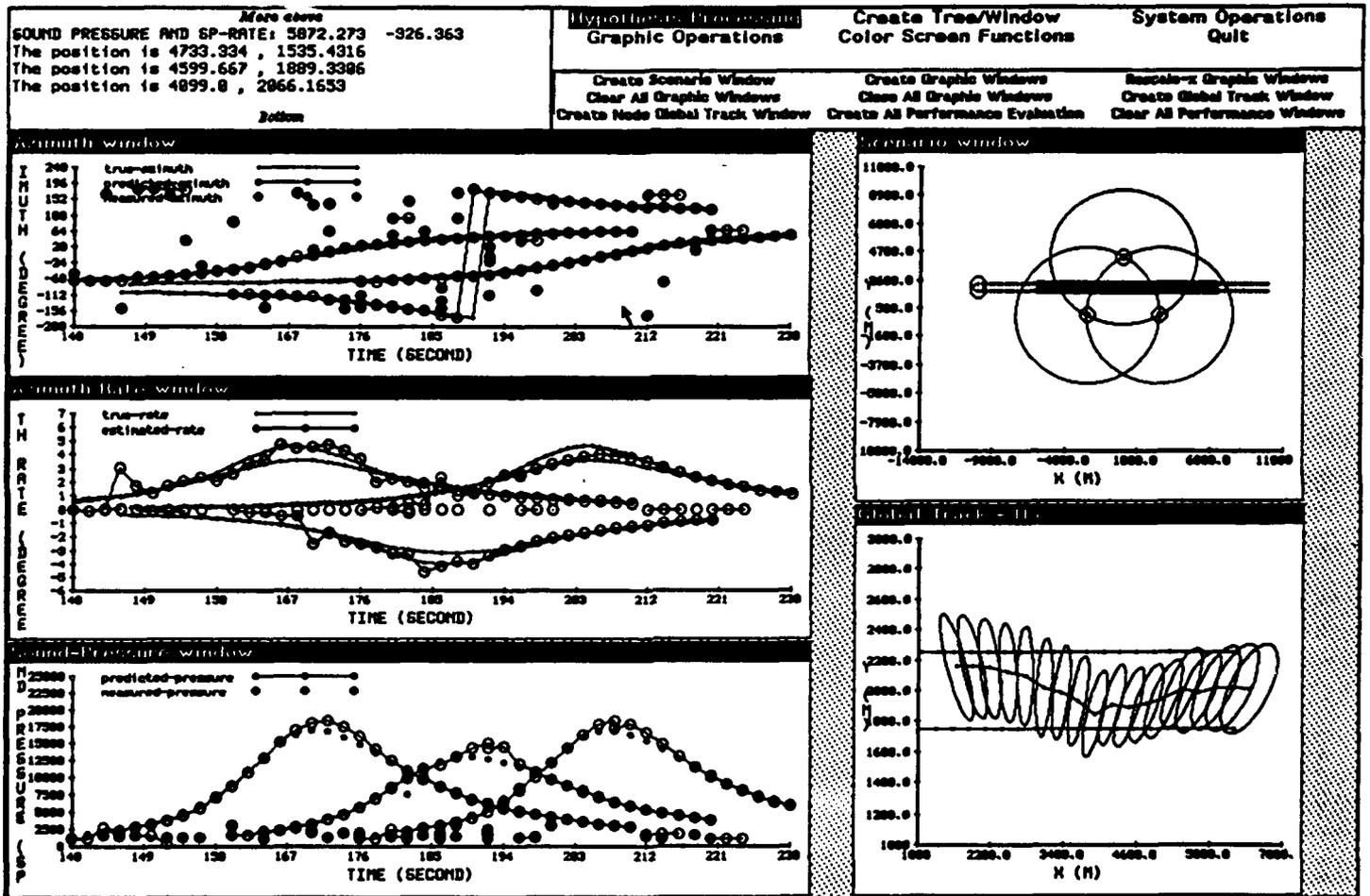


Figure 6-29: Tracking Results of Parallel Formation I (Scenario 2-3)

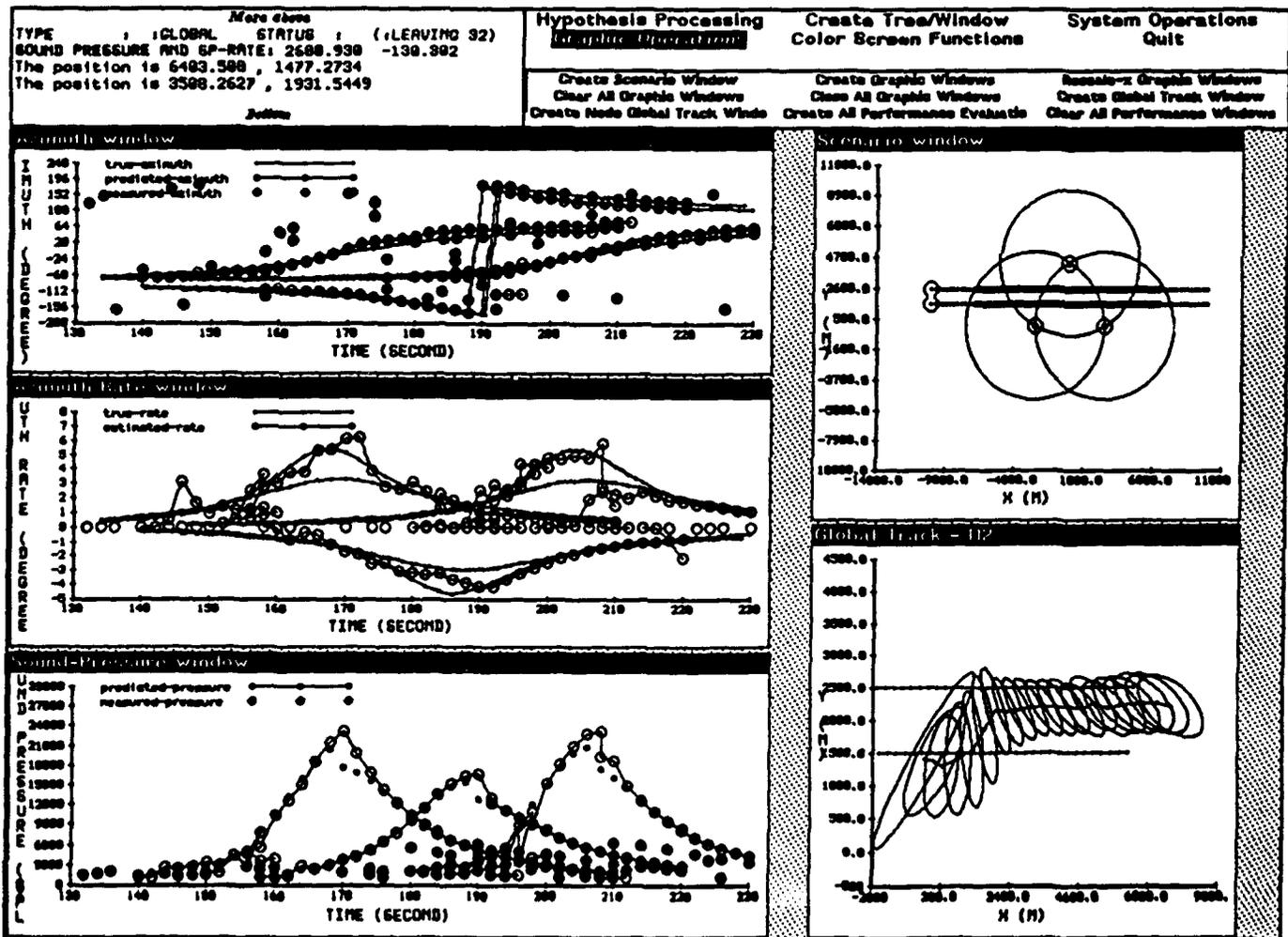


Figure 6-30: Tracking Results of Parallel Formation II (Scenario 2-4)

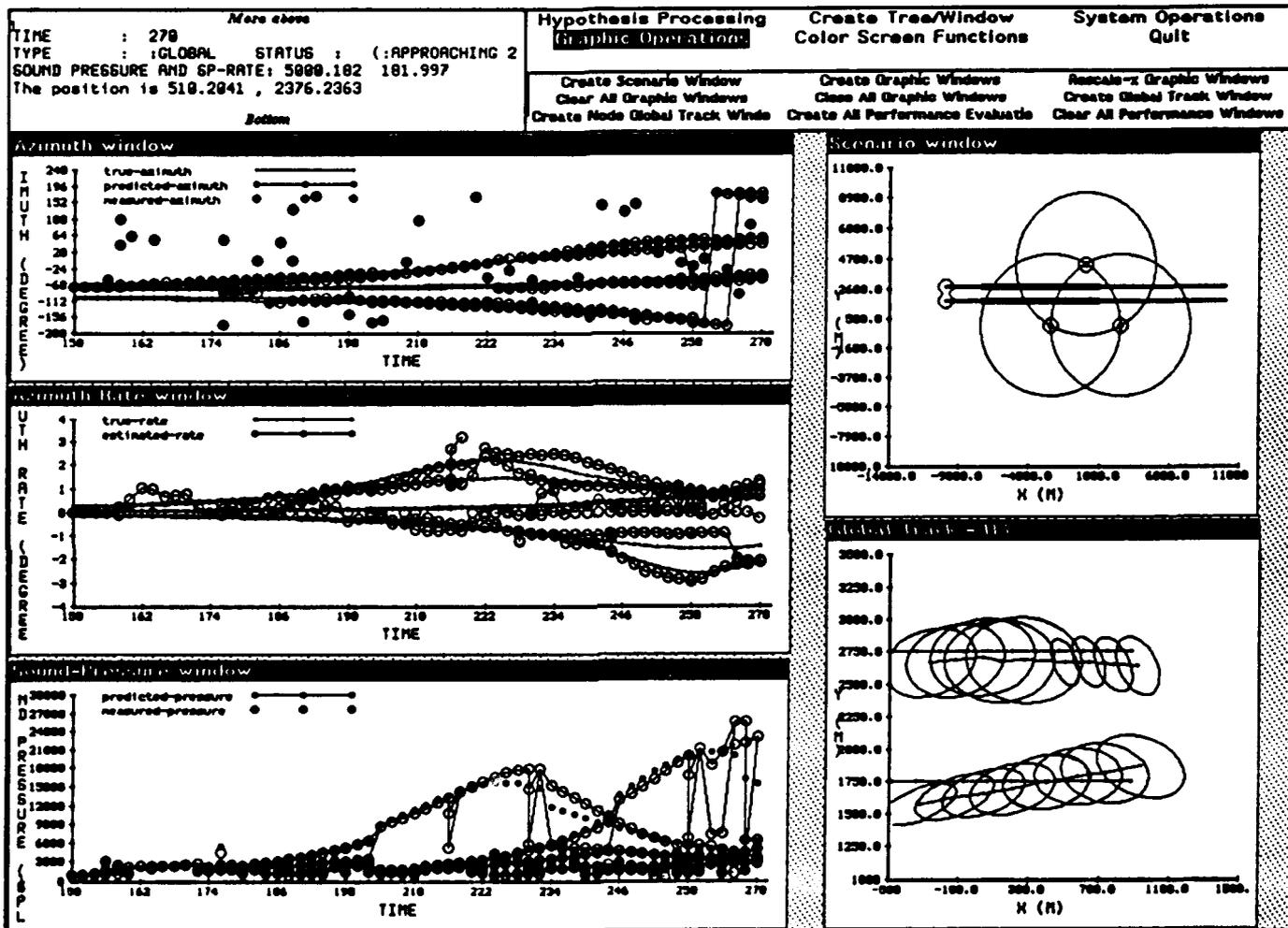


Figure 6-31: Tracking Results of Parallel Formation III (Scenario 2-5)

Table 6-6: Performance Results - Two-Targets (II)

Scenario 2-3	Node 1	Node 2	Node 3
# of Tracks	1.0	1.0	1.0
# of Missing Targets	1.0	1.0	1.0
# of False Tracks	0.0	0.0	0.0
Pos. Error(m)	568.71	527.30	306.31
Vel. Error(m/s)	21.54	10.06	3.98
Scenario 2-4	Node 1	Node 2	Node 3
# of Tracks	1.6	1.3	1.0
# of Missing Targets	1.0	1.6	1.0
# of False Tracks	0.6	0.9	0.0
Pos. Error(m)	1318.66	1700.45	1235.93
Vel. Error(m/s)	43.38	56.05	28.55
Scenario 2-5	Node 1	Node 2	Node 3
# of Tracks	2.2	2.0	2.0
# of Missing Targets	0.5	0.1	0.2
# of False Tracks	0.7	0.1	0.2
Pos. Error(m)	313.35	467.51	146.06
Vel. Error(m/s)	7.02	13.07	6.25

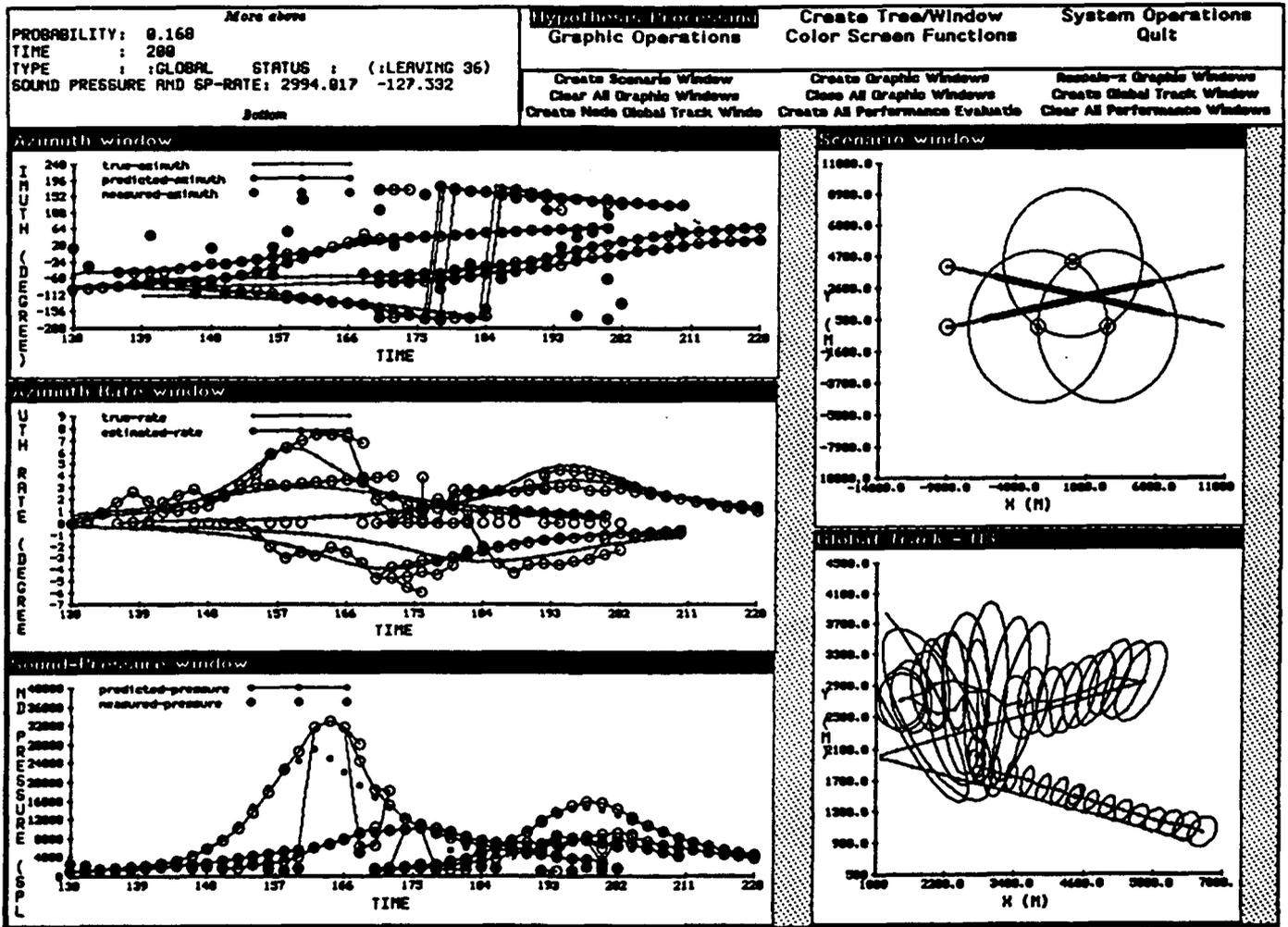


Figure 6-32: Tracking Results of Crossing Targets I (Scenario 2-6)

close. In Figure 6-33, where targets move at only half of the speed of the previous case, node 3 tracks both targets accurately throughout most of the simulation. Again, in this case, by slowing down the targets, the observability of the resolved measurements was increased and the tracking performance was better. Table 6-7 gives the performance results for the two cases.

### 6.3.1.3 Three Target Scenario

Two three target scenarios with parallel and in-line formations are given in Figure 6-34. Figure 6-35 shows the scenario for parallel formation and the tracking results from the left-top three sensors. This case is particularly difficult because during certain periods, the three targets are almost on the same azimuth relative to a sensor and generate only one merged measurement.

Finally, Figure 6-36 shows the three-target in-line formation scenario and the tracking results. This case is similar to the two-target in-line formation case, and the algorithm also performs very well here. The performance results of the two cases are summarized in Table 6-8.

## 6.3.2 Simulation with Real Data

The real data provided by Lincoln Lab. were generated from two sensors and one constant-speed target. The target speed is about 0.2 Mach and it passes by very near to one of the sensors (see Figure 6-37). The original raw data were extremely noisy and the false measurements were not quite independent from scan to scan. Figures 6-38 and 6-39 show the raw measurements (after signal processing) from sensors 1 and 2 respectively. There are several persistent measurement strings; of these, only one from each sensor comes from the target and all the others are false measurements.

To avoid extensive processing in the dense non-independent false measurement environment, two pre-processing strategies were employed. One was based on using a threshold and the other an adaptive threshold. In the first case, all the measurements were passed through a sound pressure threshold filter, and only those above the threshold (300 in this case) were retained. The filtered measurements were then sent to the tracking algorithm. In the second case, at each scan the energy (sound pressure) of all measurements were sorted and only the top 95%

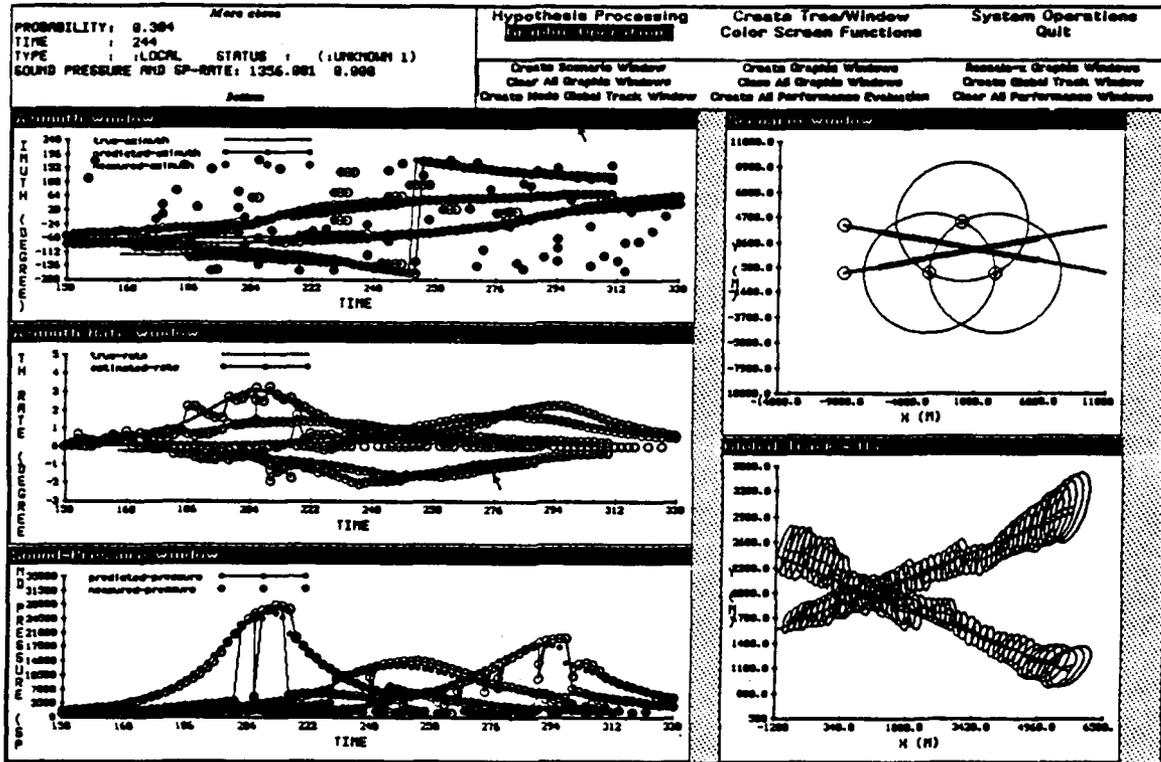


Figure 6-33: Tracking Results of Crossing Targets II (Scenario 2-7)

Table 6-7: Performance Results - Two-Target (III)

Scenario 2-6	Node 1	Node 2	Node 3
# of Tracks	1.1	2.0	2.0
# of Missing Targets	1.0	1.0	0.0
# of False Tracks	0.1	1.0	0.0
Pos. Error(m)	411.62	336.70	162.88
Vel. Error(m/s)	20.56	12.62	7.74
Scenario 2-7	Node 1	Node 2	Node 3
# of Tracks	2.0	2.0	2.0
# of Missing Targets	0.0	0.0	0.0
# of False Tracks	0.0	0.0	0.0
Pos. Error(m)	149.91	282.67	211.02
Vel. Error(m/s)	3.57	6.68	3.83

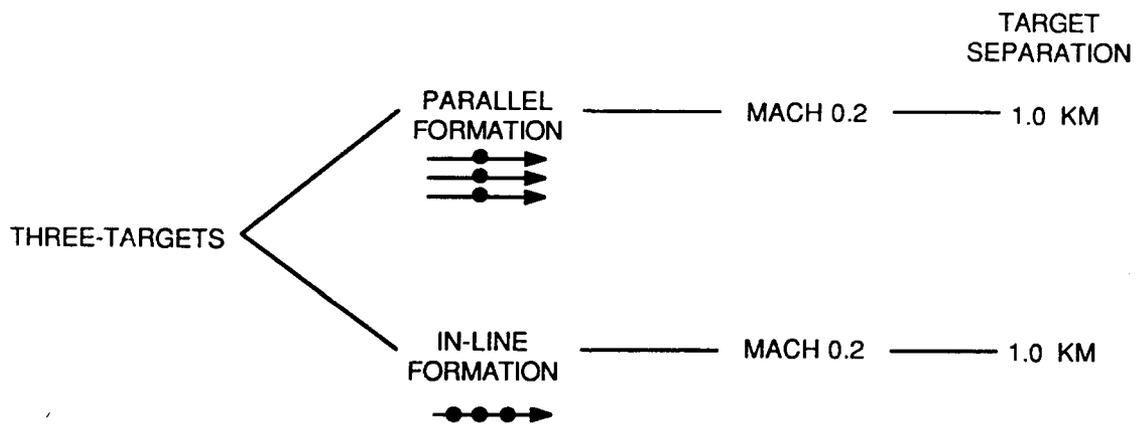


Figure 6-34: Three-Target Scenarios

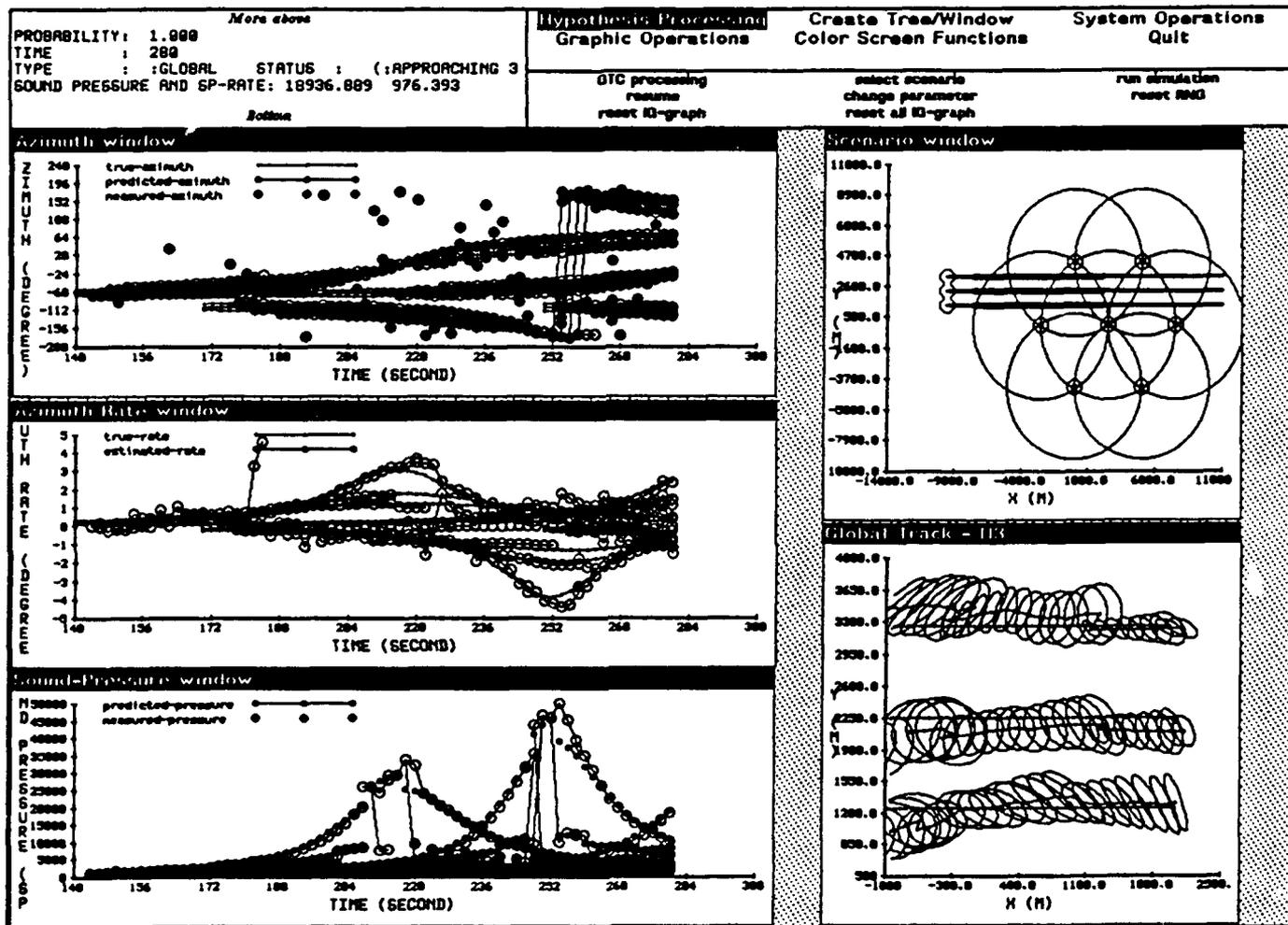


Figure 6-35: Tracking Results of Three-Target Case I (Scenario 2-8)

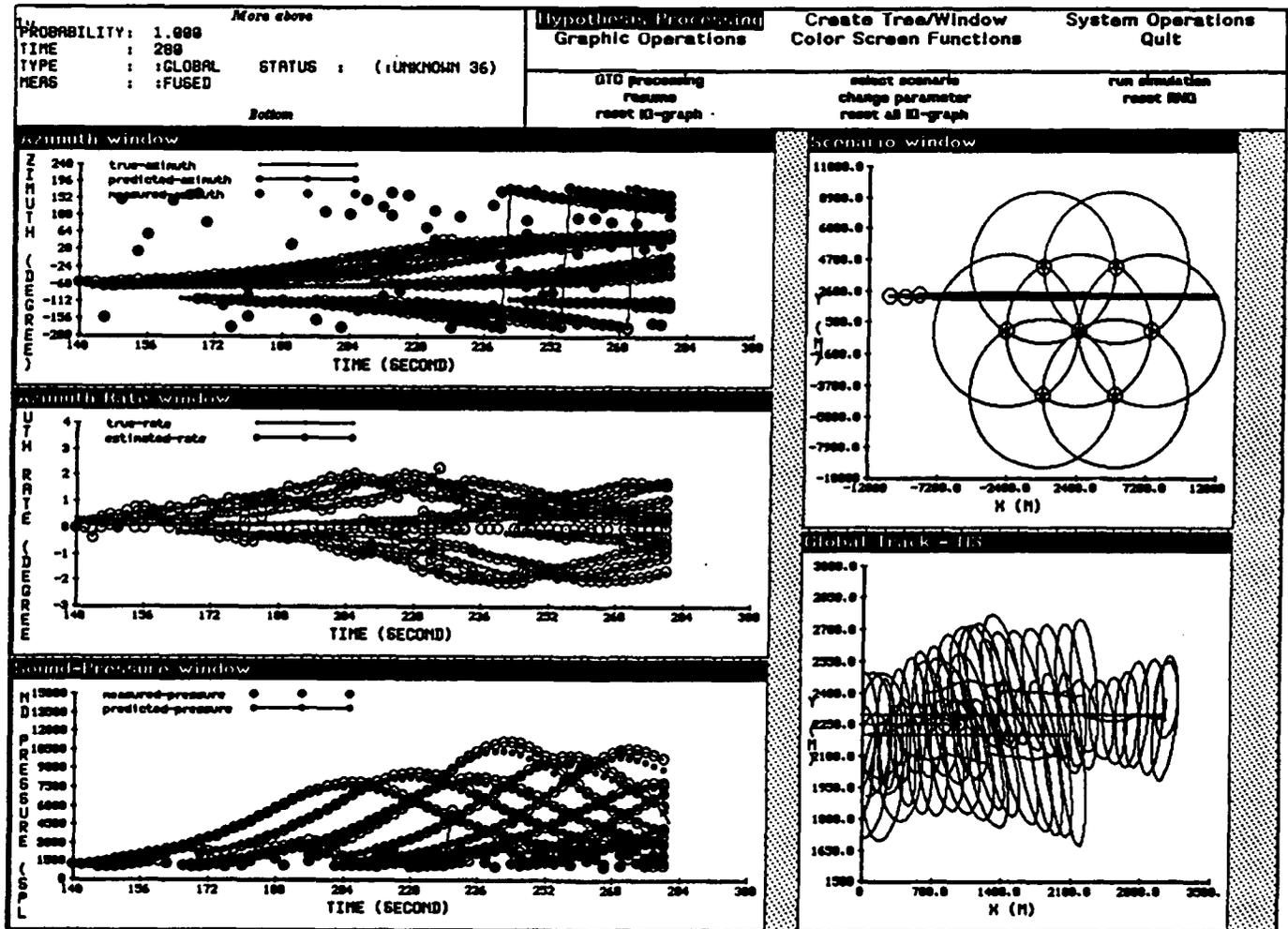


Figure 6-36: Tracking Results of Three-Target Case II (Scenario 2-9)

Table 6-8: Performance Results - Three-Target Case

Scenario 3-1	Node 1	Node 2	Node 3
# of Tracks	2.5	3.0	3.0
# of Missing Targets	0.5	0.0	0.3
# of False Tracks	0.0	0.0	0.3
Pos. Error(m)	643.92	477.97	307.25
Vel. Error(m/s)	16.41	17.89	14.47
Scenario 3-2	Node 1	Node 2	Node 3
# of Tracks	3.0	3.0	3.0
# of Missing Targets	0.0	0.0	0.0
# of False Tracks	0.0	0.0	0.0
Pos. Error(m)	37.39	38.30	61.86
Vel. Error(m/s)	1.48	1.72	2.18

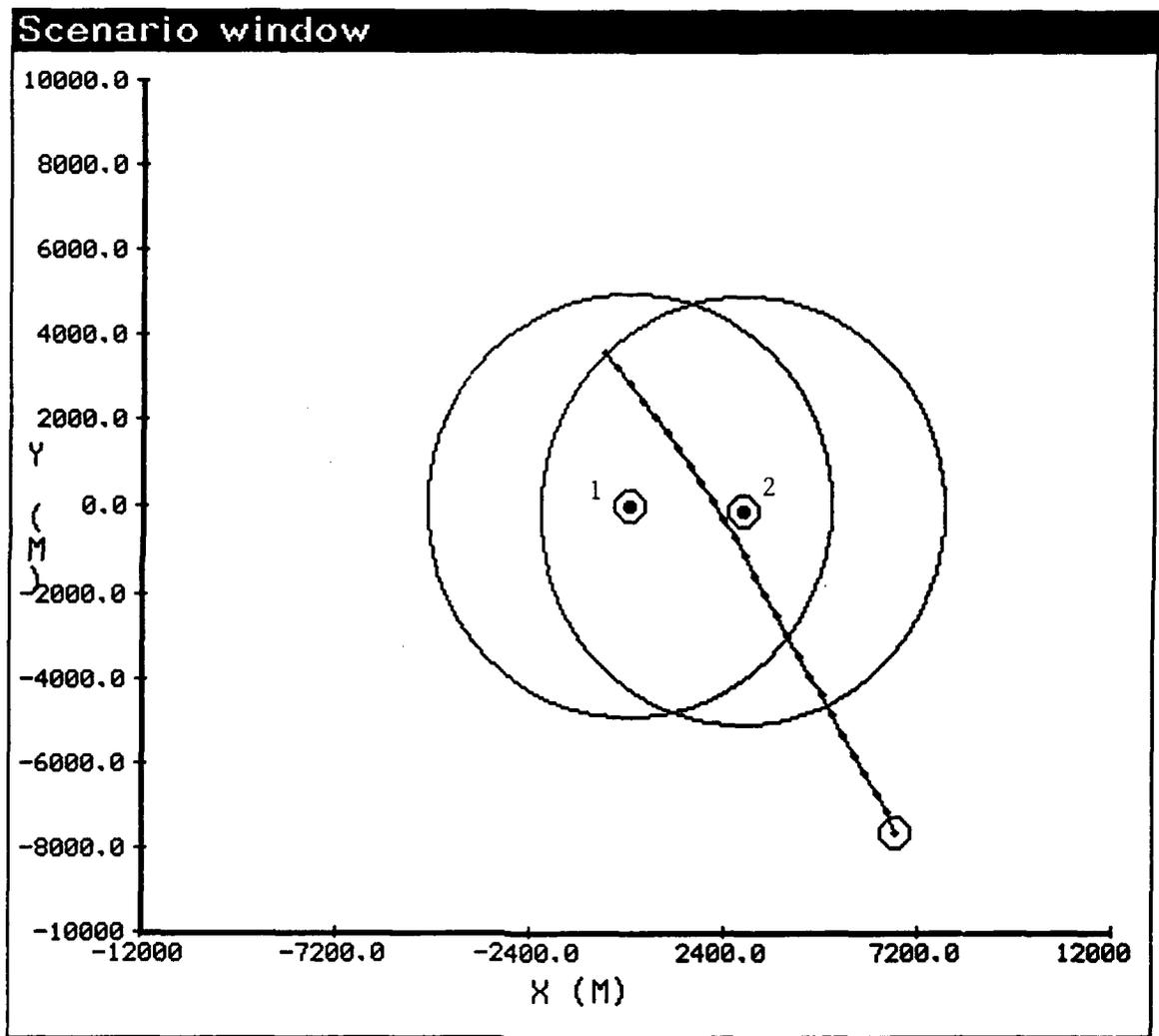


Figure 6-37: Real Data Scenario (Scenario 2-10)

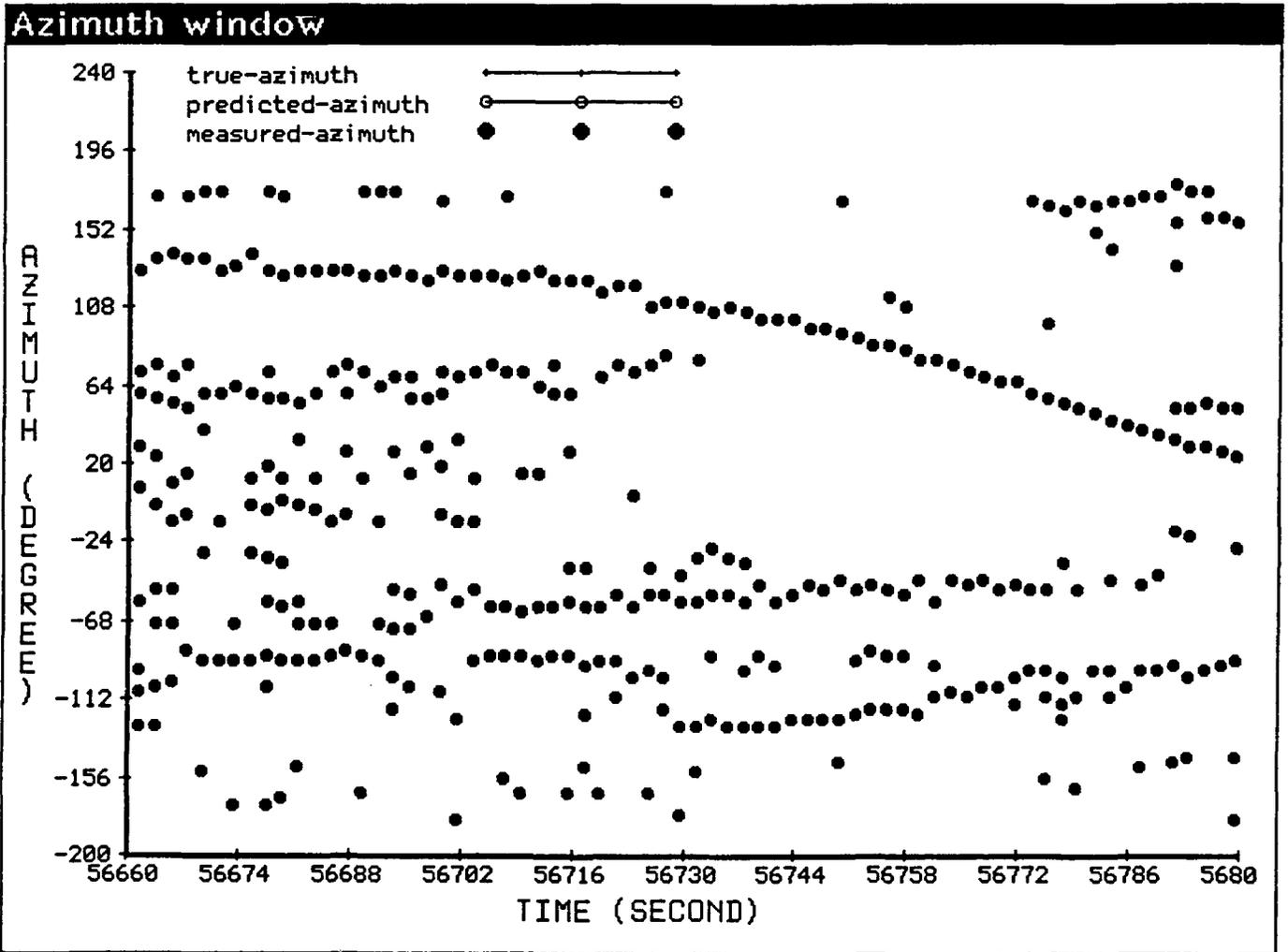


Figure 6-38: Raw Data from Sensor 1

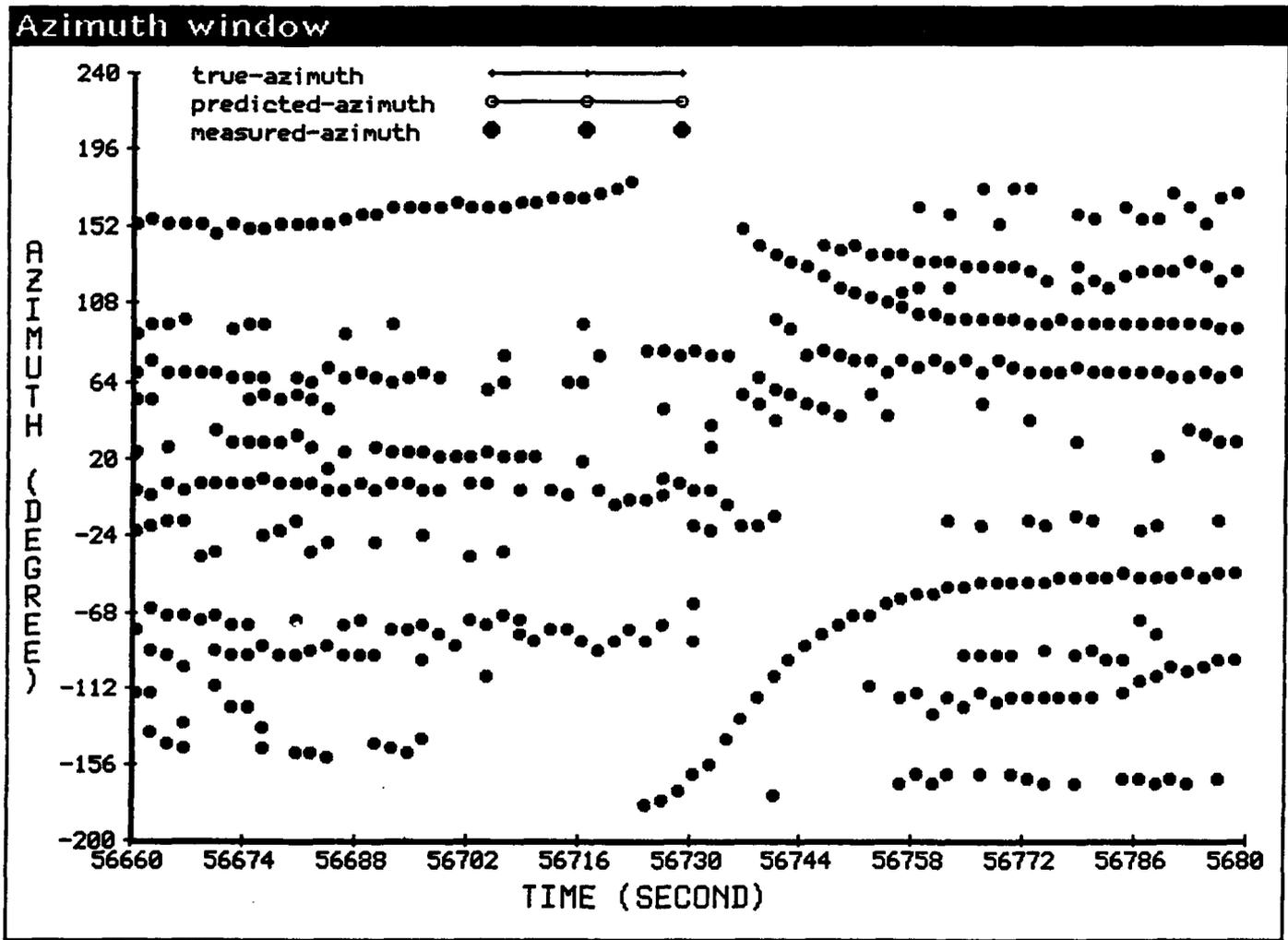


Figure 6-39: Raw Data from Sensor 2

of the measurements were kept. The filtered measurements for both cases are given in Figures 6-40 to 6-43.

The tracking results using both types of pre-processing are given in Figures 6-44 and 6-45. In the second case, since most of the false measurements were eliminated, it was relatively easy to track the target and almost no false tracks were generated. On the other hand, there were still a few false measurement strings left in the first case. Each sensor established several local azimuth tracks based on those measurements. Many global tracks were initiated when the local tracks of two sensors were first communicated. However, except for the true global track, most of the false global tracks vanished after a few scans. The false global tracks were eliminated either due to the lack of support from local measurements or conflict with the other sensor during later communications. Finally, at the end of the simulation, only the correct global track survived for each node. Table 6-9 gives the performance results of the two cases.

#### **6.4 MONTE CARLO RESULTS**

In the previous section, we presented simulation results of single runs for 20 scenarios. Monte Carlo simulations were conducted for several scenarios to obtain a more reliable evaluation of the tracking performance. For each scenario, selected parameters were varied to study the sensitivity of the performance to the parameters. The performance curves were then obtained.

Three different scenarios were chosen for the Monte-Carlo simulations. They correspond to a single constant speed target, a single maneuvering target, and two targets in a line formation. Two parameters were varied in each case. For each set of parameter values, ten Monte-Carlo runs were simulated. The performance results of each run were calculated by averaging the results of the last five scans. The overall performance of each case was then obtained by averaging the results of the ten runs.

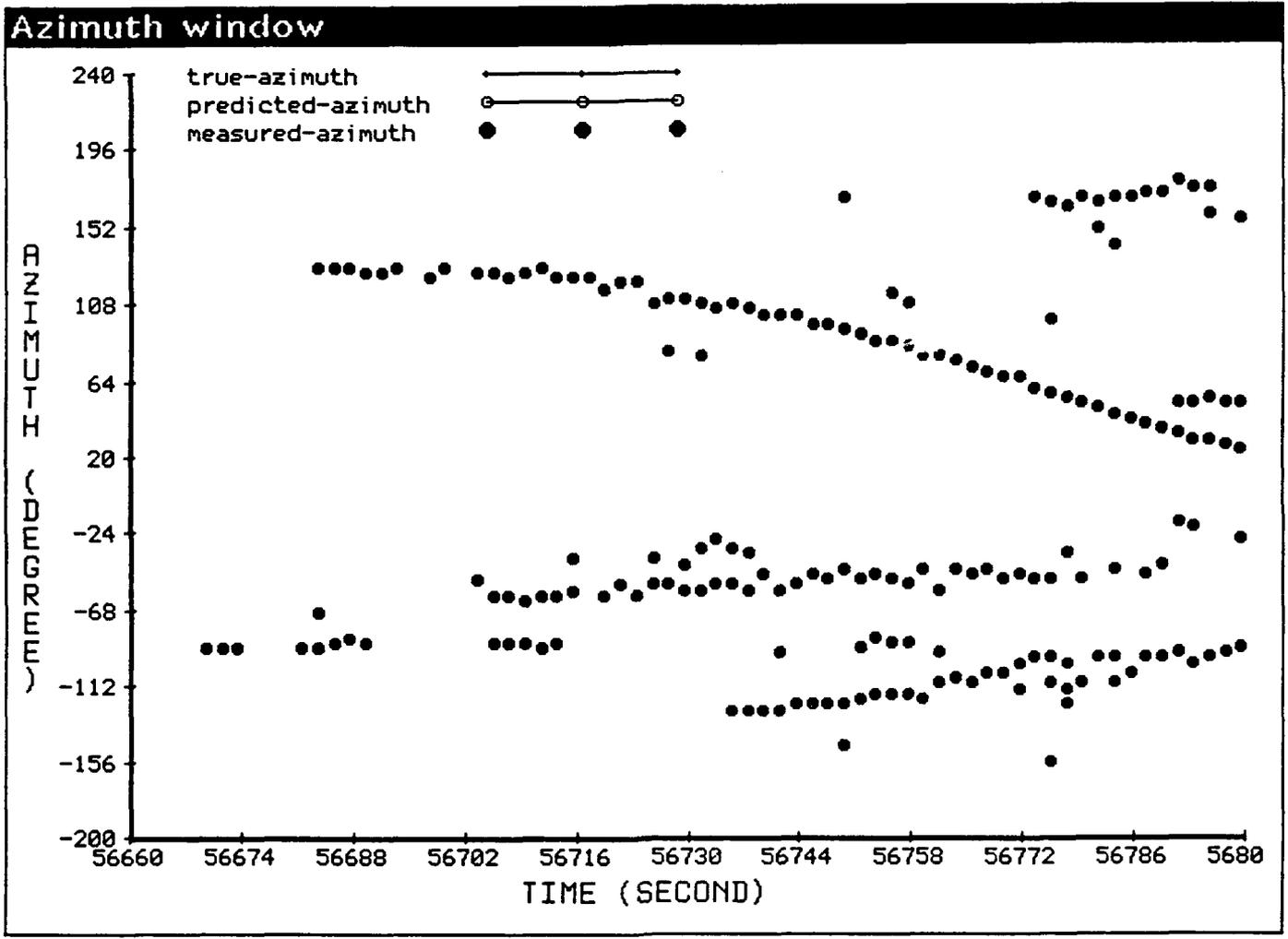


Figure 6-40: Filtered Data from Sensor 1 - Fixed Threshold

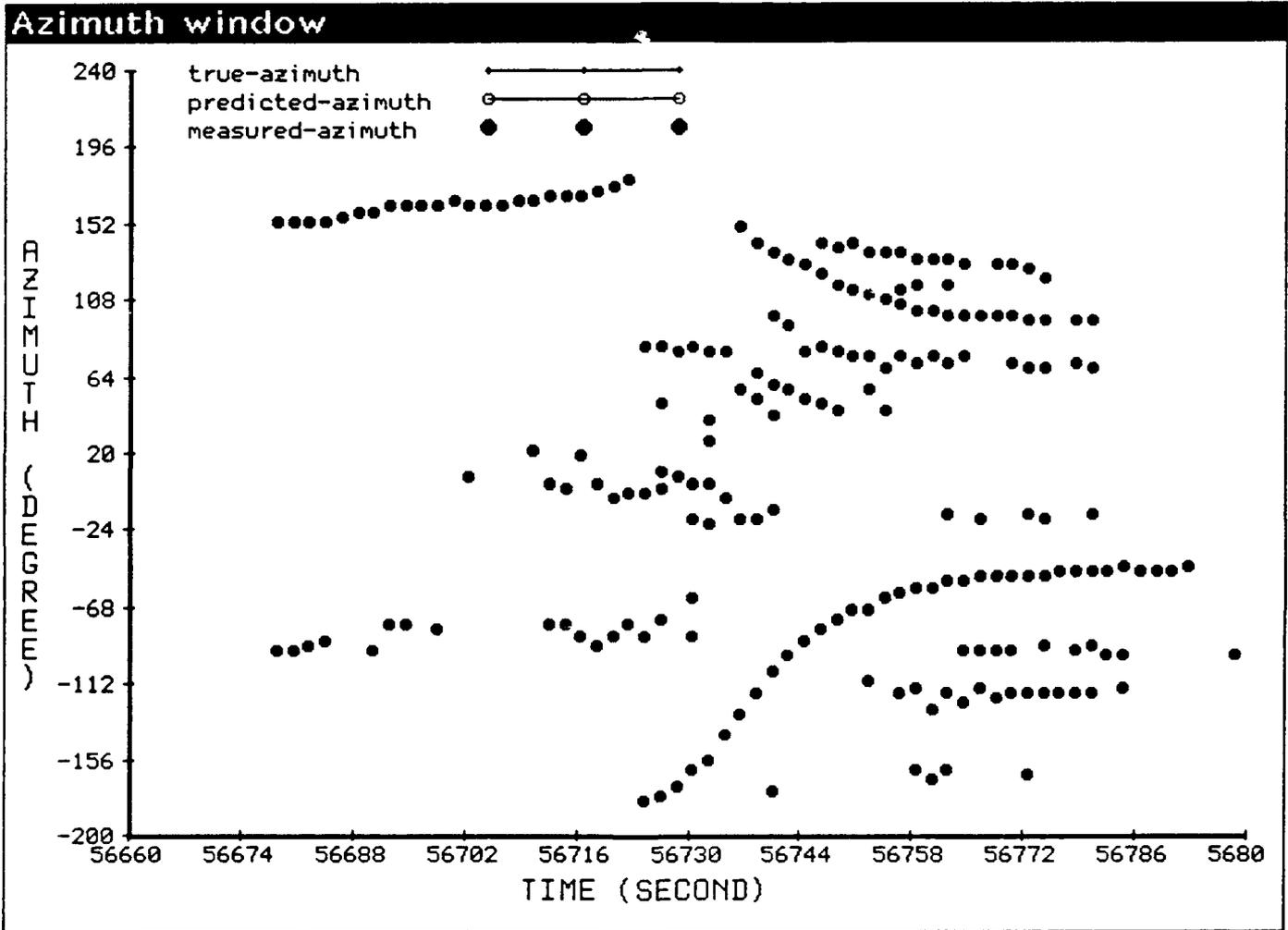


Figure 6-41: Filtered Data from Sensor 2 - Fixed Threshold

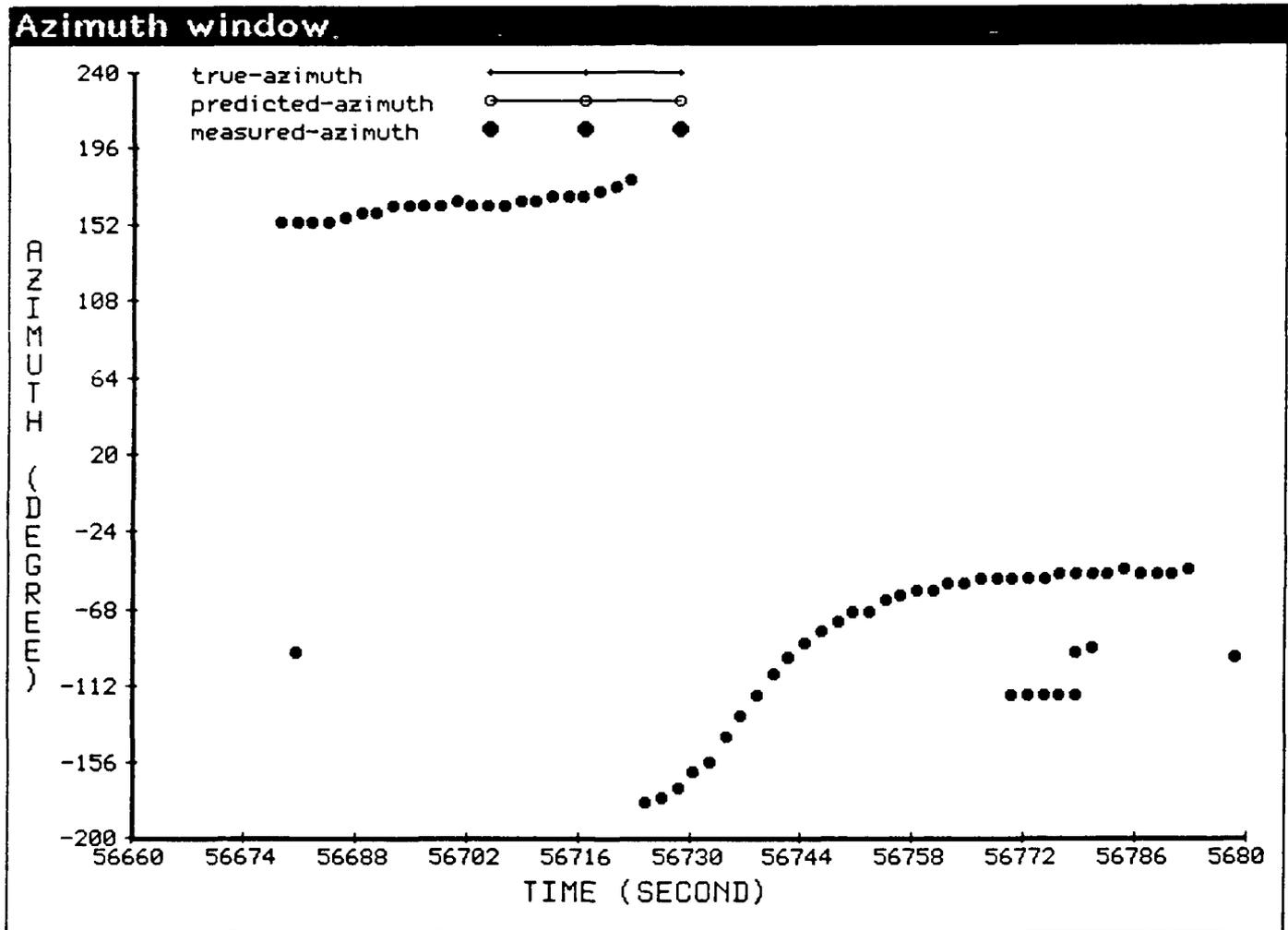


Figure 6-42: Filtered Data from Sensor 1 - Adaptive Threshold

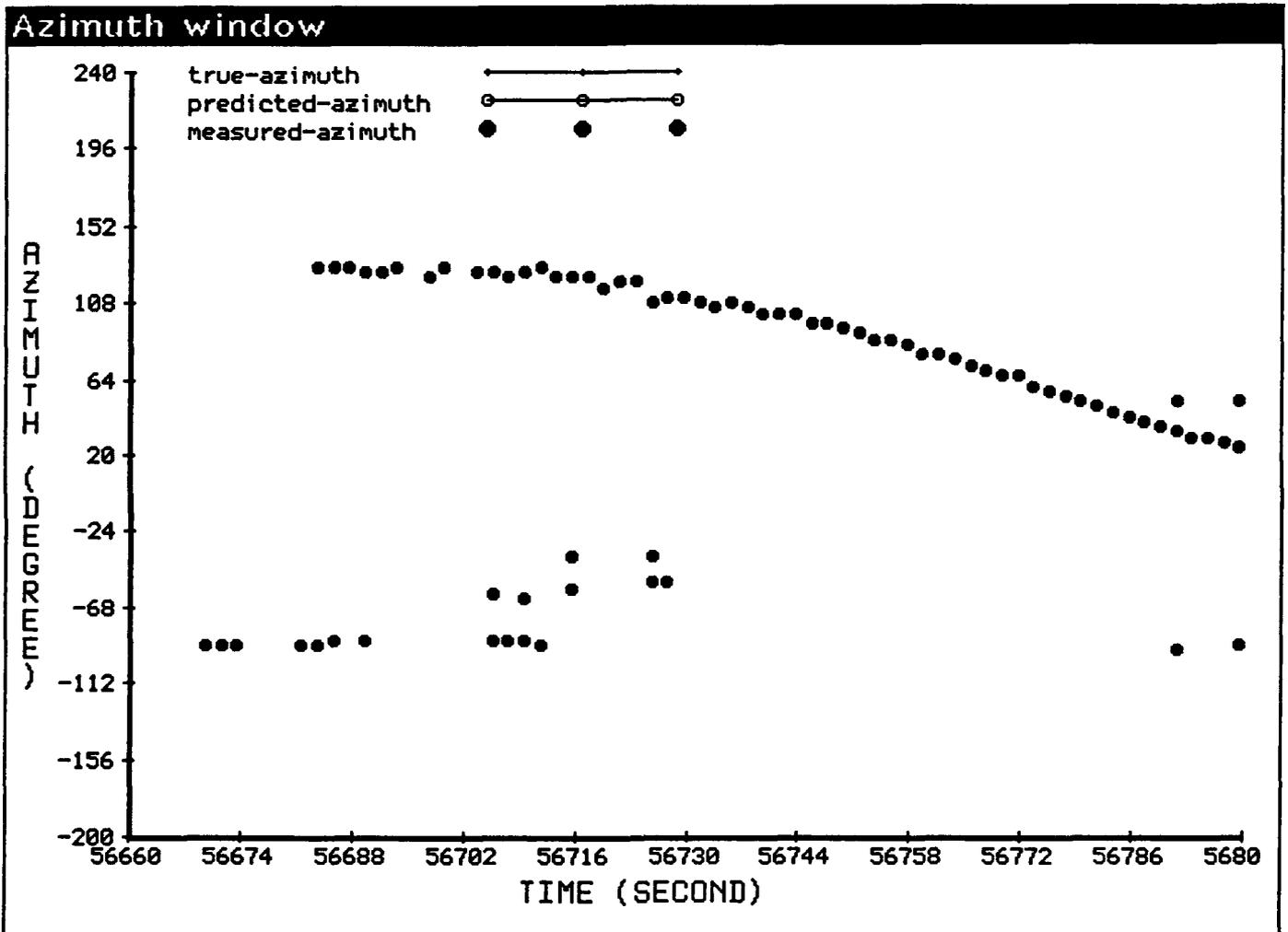


Figure 6-43: Filtered Data from Sensor 2 - Adaptive Threshold

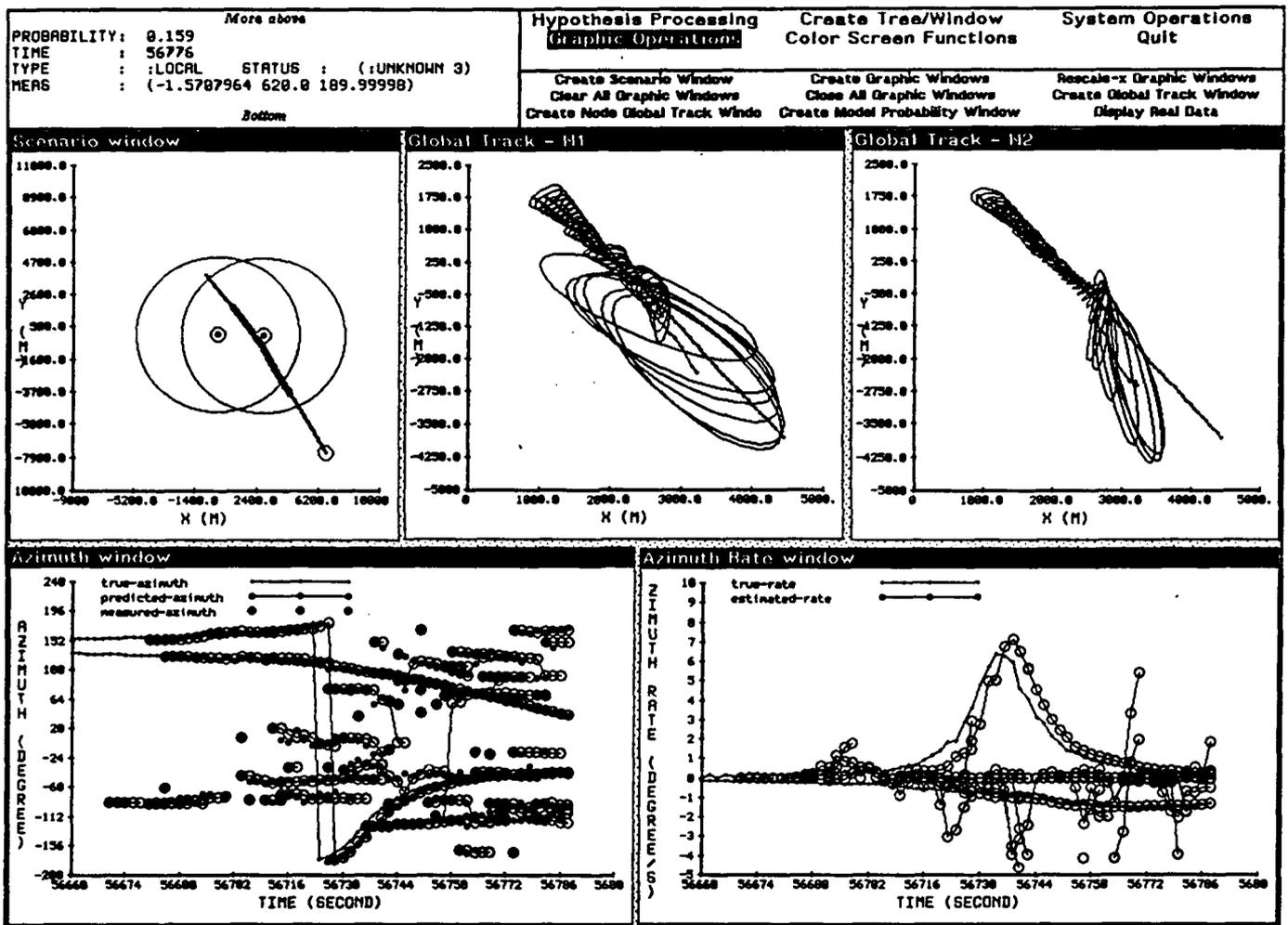


Figure 6-44: Tracking Results of Real Data I

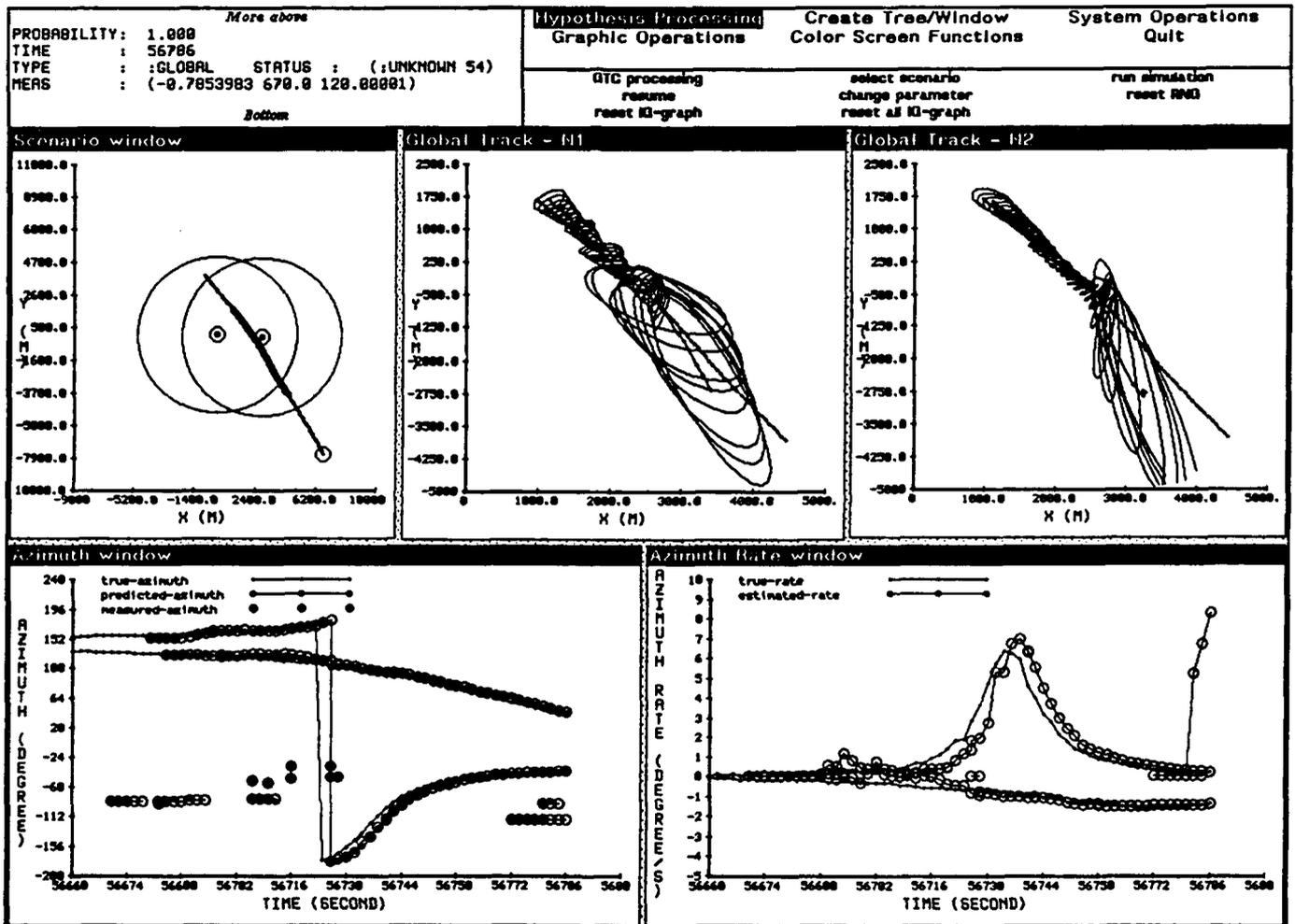


Figure 6-45: Tracking Results of Real Data II

Table 6-9: Performance Results of Real Data Cases

Fixed-Threshold	Node 1	Node 2
# of Tracks	3.9	4.5
# of Missing Targets	0.0	0.0
# of False Tracks	2.9	3.5
Pos. Error(m)	247.75	228.16
Vel. Error(m/s)	21.45	21.95
Adaptive-Threshold	Node 1	Node 2
# of Tracks	1.0	1.9
# of Missing Targets	0.0	0.0
# of False Tracks	0.0	0.9
Pos. Error(m)	118.89	98.55
Vel. Error(m/s)	7.10	7.86

#### 6.4.1 Scenario I (Single Constant Speed Target)

This scenario is similar to the one in Figure 6-16. Two parameters, the false alarm rate and frequency of communication, were chosen to study sensitivity of the performance. Four performance indices, including number of false tracks, number of missing targets, position error, and velocity error, were used to measure the performance. The results of a typical sample run are given in Figure 6-46,

where the false alarm rate was 0.5 and the minimum communication period was 5. Figures 6-47 and 6-48 show the curves of the average position error and number of false tracks versus the parameter values. As one can see, within given ranges, both the false alarm rate and the frequency of communication do not affect the performance significantly. However, when the false alarm rate is too high (above 1.5) or the communication period is too long (above 10), the performance starts to degrade.

#### **6.4.2 Scenario II (Single Maneuvering Target)**

This scenario is similar to the one given in Figure 6-25. However, the target speed is set to be 0.3 Mach. In this scenario, only the sensor resolution will be varied. Performance measures similar to the previous scenario were used to evaluate the results. Figure 6-49 shows the results of a typical run with a communication period of 5 and a sensor resolution of 10 degrees. The performance curves in Figure 6-50 show a clear relationship between the sensor resolution and the tracking accuracy. When the sensor resolution increases to 15 degrees, the average number of missing targets also starts to increase.

#### **6.4.3 Scenario III (Two Targets in Line Formation)**

The two targets flying in a formation are considered in this case. In this scenario the two parameters to be varied are target separation and sensor resolution. Figure 6-51 shows the results of a typical run with a communication period of 5 and a sensor resolution of 10 degrees. The performance curves for the average positional error and number of missing targets are given in Figures 6-52 and 6-53.

In the first set of results (Figure 6-52), the sensor resolution was fixed to be 10 degrees, and the target separation was varied from 0.5 km to 1.5 km. While on the average one target was missed when the target separation was 0.5 km, both targets were tracked when the target separation increased to about 1.0 km. In the second set of curves, we fixed the target separation to be 1.0 km and varied the sensor resolution from 5 degrees to 20 degrees. Figure 6-53 shows that both targets were tracked when the sensor resolution was below 10 degrees. When the sensor resolution is greater than 15 degrees, the algorithm failed to distinguish between the two separate targets.

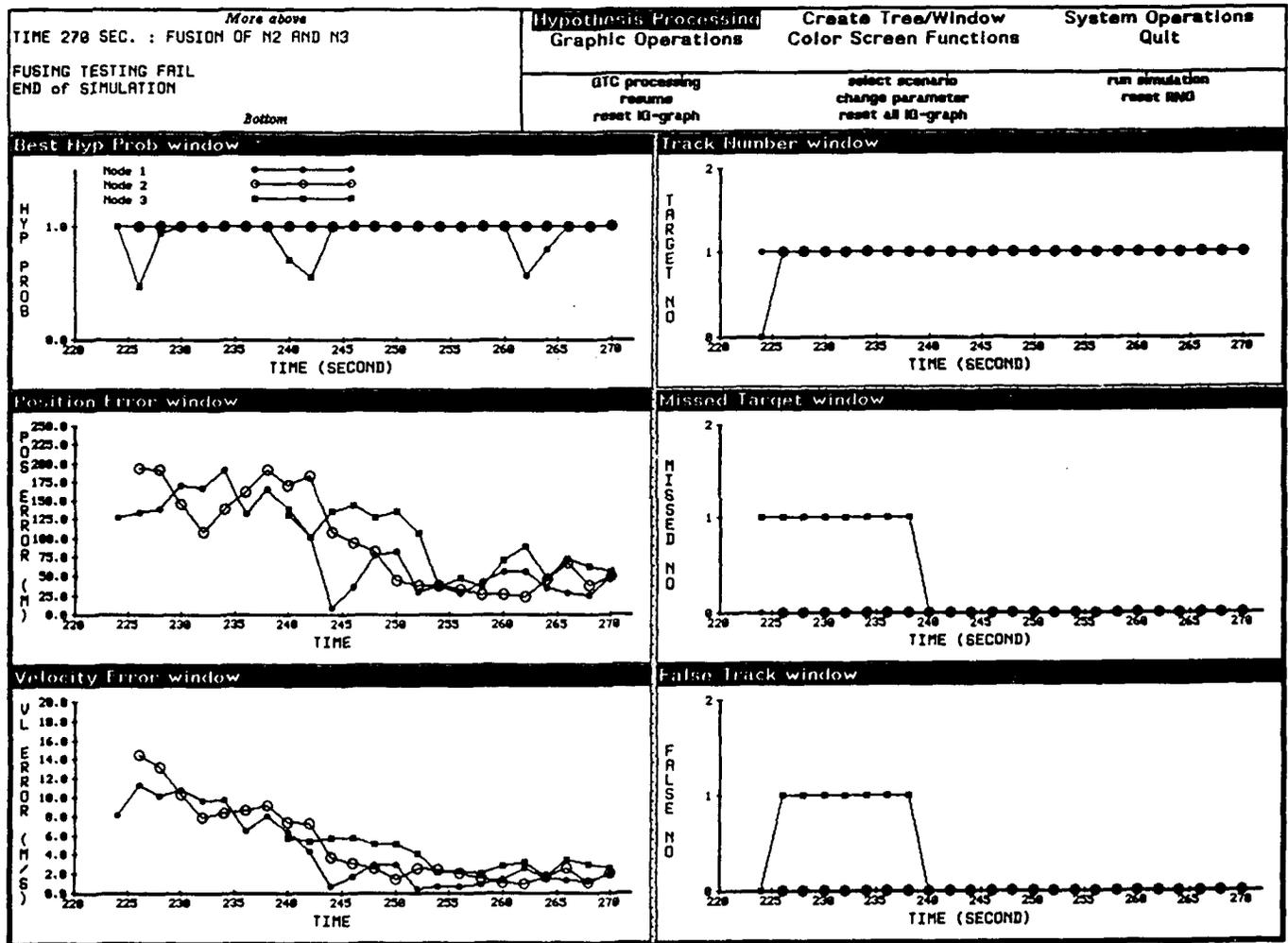
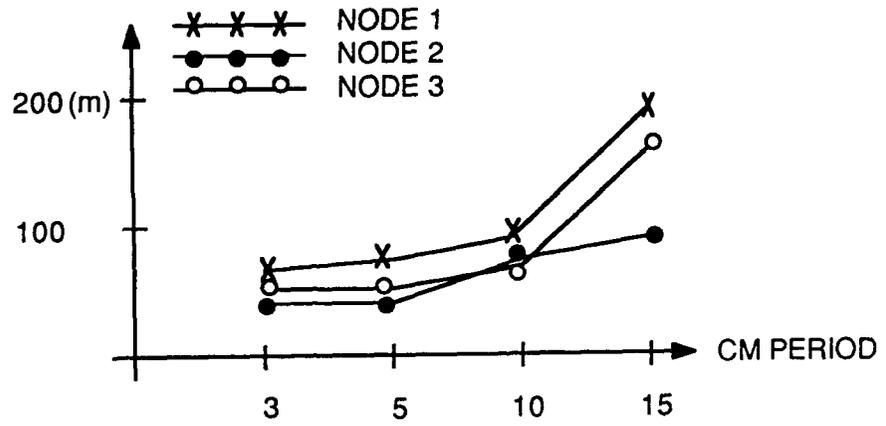


Figure 6-46: Typical Performance Results for Case I

AVERAGE POSITION ERROR



AVERAGE NUMBER OF FALSE TRACKS

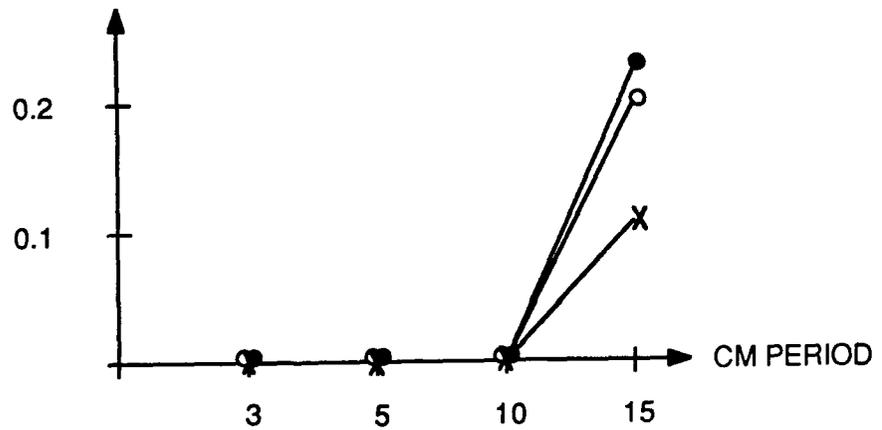
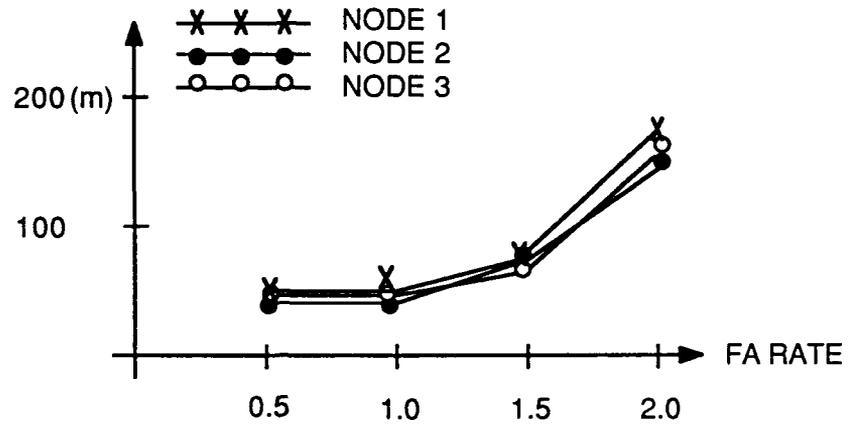


Figure 6-47: One-Target Constant Speed Case with False Rate = 0.5

AVERAGE POSITION ERROR



AVERAGE NUMBER OF FALSE TRACKS

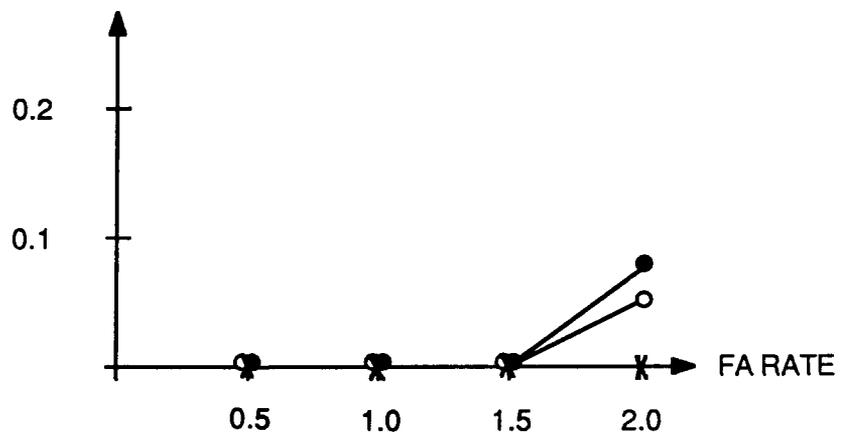


Figure 6-48: One-Target Constant Speed Case with Comm. Period = 5

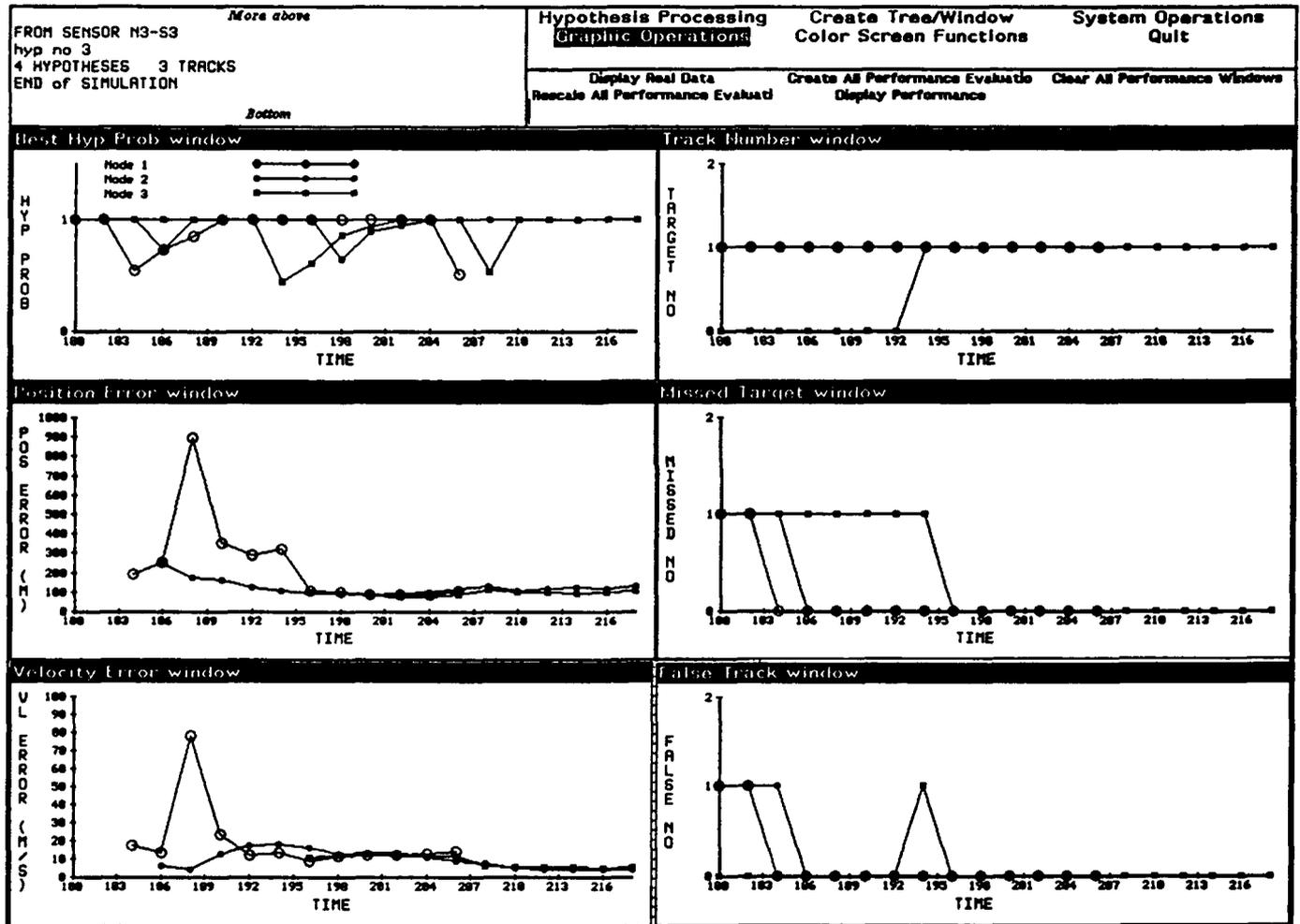


Figure 6-49: Typical Performance Results for Case II

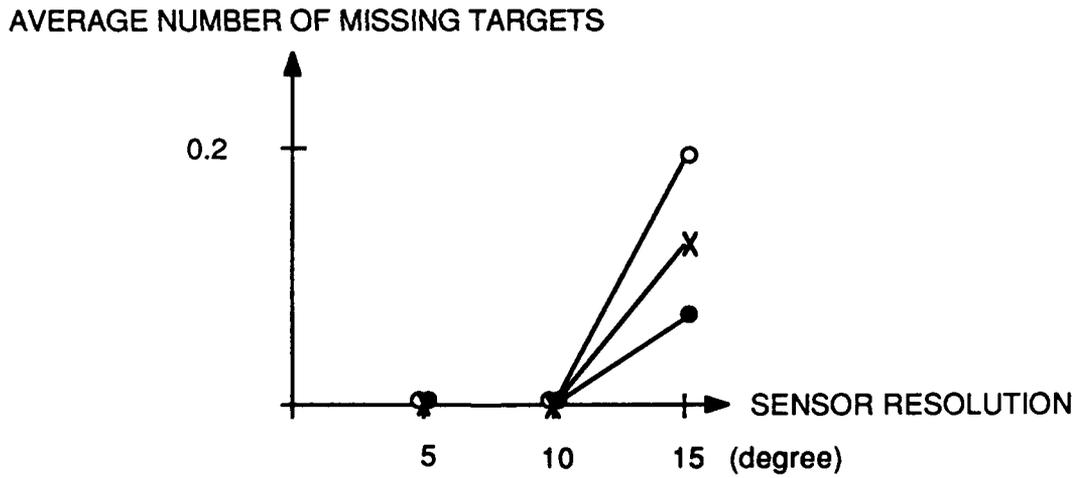
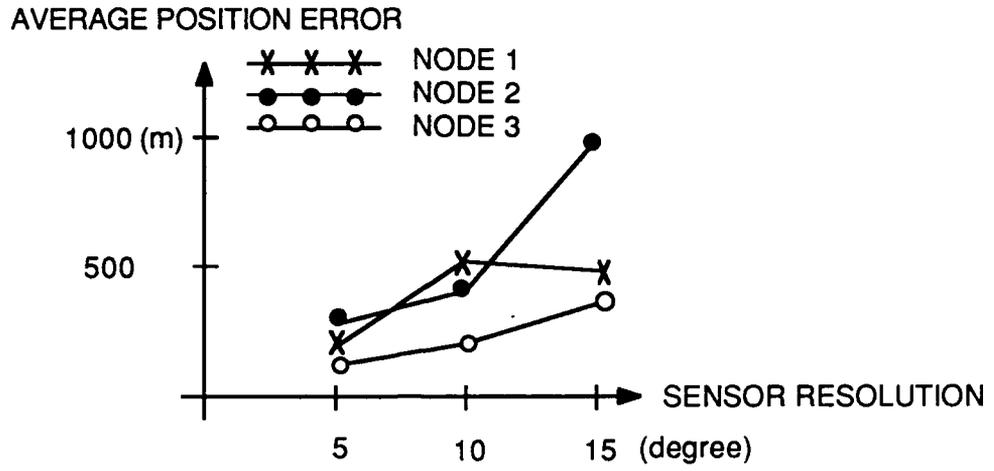


Figure 6-50: One Maneuvering Target Case with Sensor Resolution = 10 Degrees

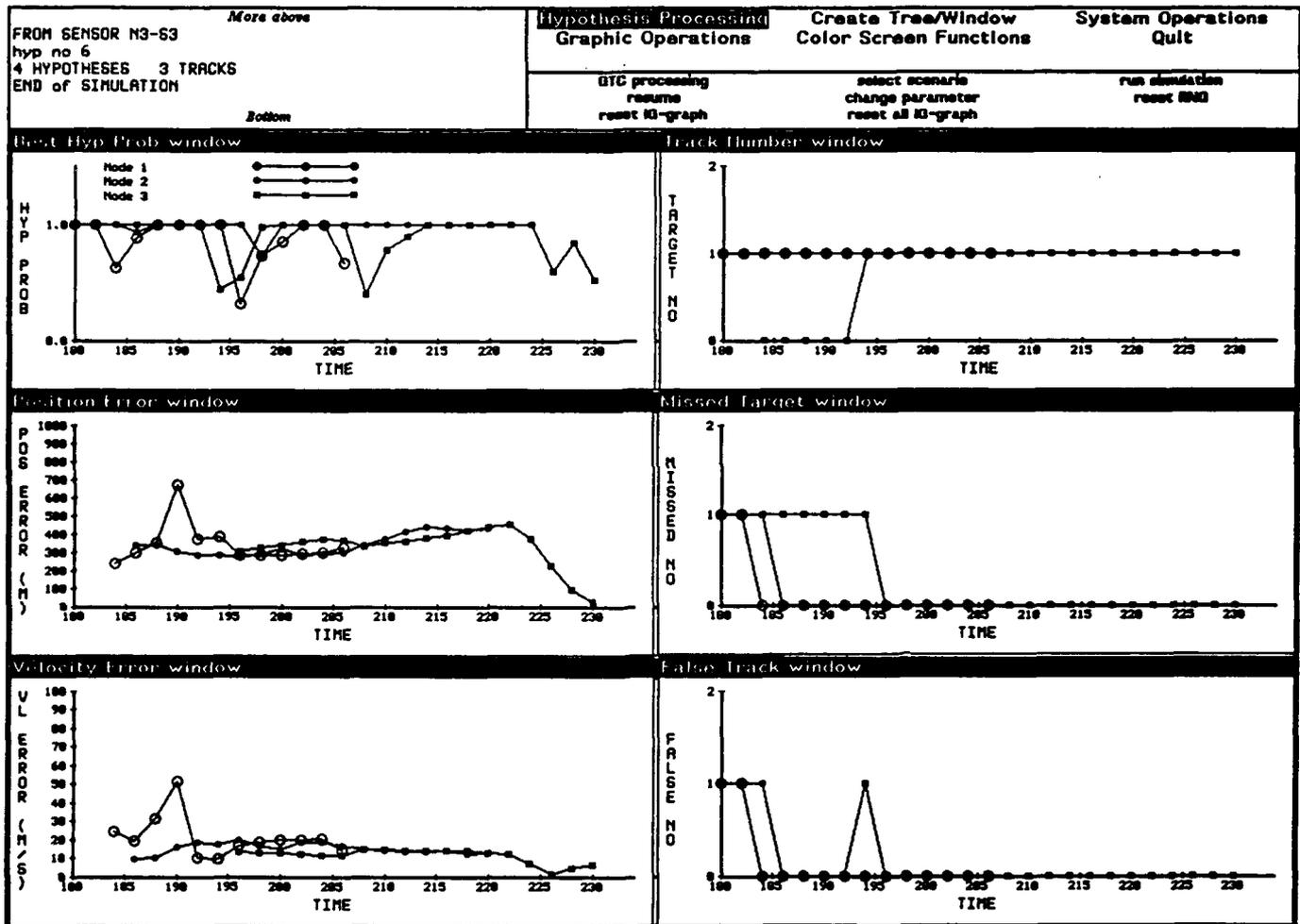
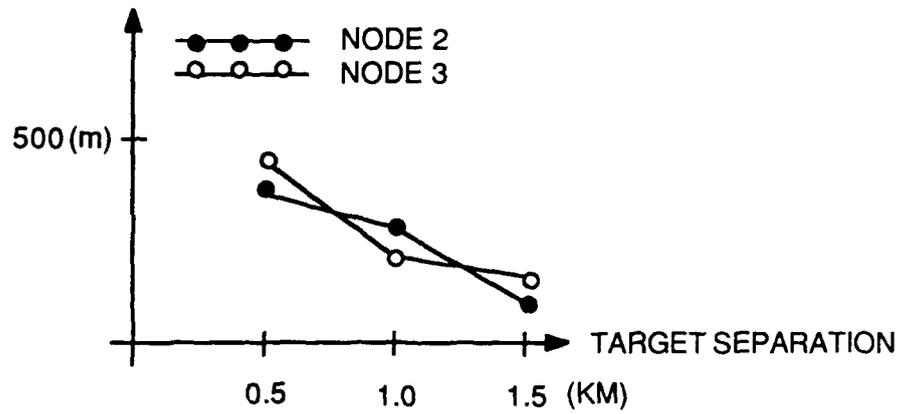


Figure 6-51: Typical Performance Results for Case III

AVERAGE POSITION ERROR



AVERAGE NUMBER OF MISSING TARGETS

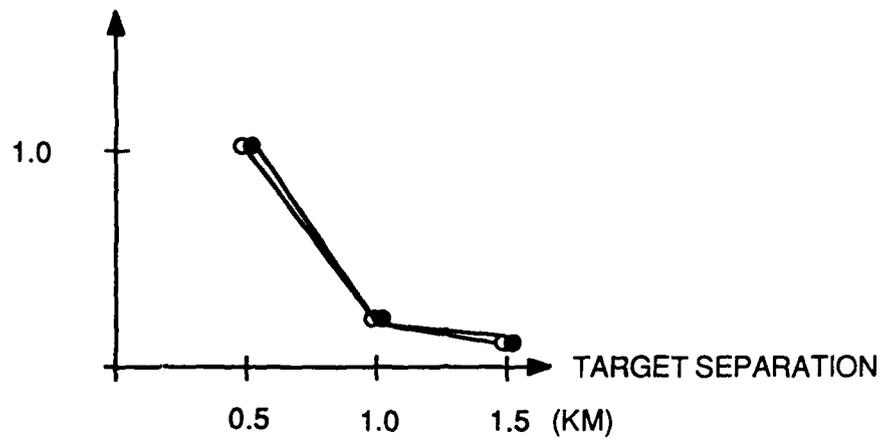
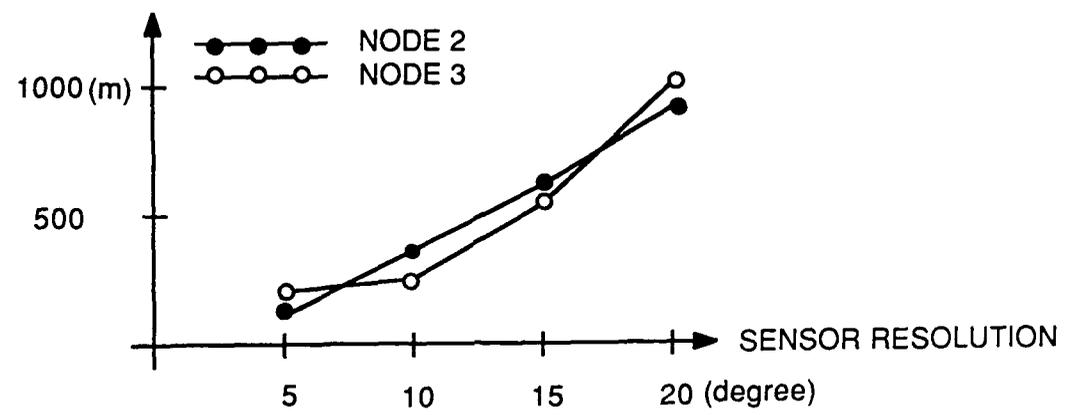


Figure 6-52: Two-Target In-Line Formation Case with Sensor Resolution 10 Degrees

AVERAGE POSITION ERROR



AVERAGE NUMBER OF MISSING TARGETS

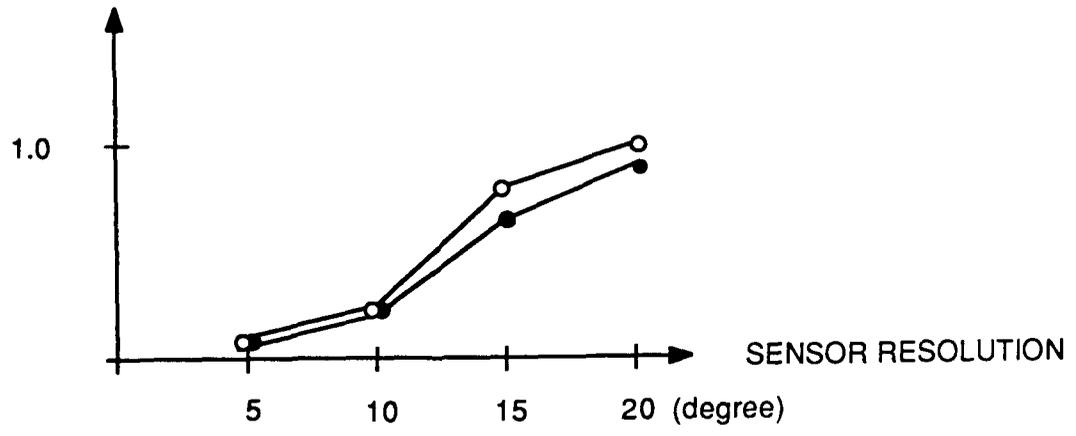


Figure 6-53: Two-Target In-Line Formation Case with Target Separation 1.0 KM

## 7. CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH

This section contains some conclusions and suggestions for future research.

### 7.1 CONCLUSIONS

The main conclusions of this research are that the general distributed tracking algorithms developed in previous DSN projects can be used for tracking air targets with acoustic sensors. Furthermore, simulation results indicated that acceptable performance can be achieved by a DSN of cooperating nodes.

In earlier DSN projects, we developed a general approach for the distributed tracking and classification of multiple targets. The approach is based on general target and sensor models and retains multiple hypotheses when the situation is unclear.

This project dealt with the tracking of multiple air targets in a DSN of acoustic sensors. Although several modifications had to be made due to the special characteristics of acoustic sensors, the basic philosophy and structure of the general algorithm was still applicable. The information fusion algorithm was modified to reflect the fact that each node may have two types of tracks, local tracks initiated locally and global tracks initiated by two cooperating nodes.

In addition to introducing local and global tracks, the other major modification was needed because of possibly unresolved measurements in the local azimuth tracking due to the poor resolution of the acoustic sensors. By considering the mechanism of merging measurements, tracks can be made to maintain relatively accurate target state estimates. The algorithm of processing merged measurements for the global tracks is similar to those for local tracks and similar results were obtained.

To establish global tracks, the target position and velocity must be estimated from two local tracks based on the acoustic azimuths and derivatives. It was found that the accuracy of such global track initiation process is very sensitive to the accuracy of the azimuth rate estimation. Thus it is very important to maintain the local tracks as accurately as possible during the periods when close

azimuth measurements from two targets merge.

Twenty different scenarios ranging from one to three targets and three to seven sensors were chosen for simulation. In the single target case, both constant speed and maneuvering targets were considered. Other scenarios included a high speed target and a target that flies very close to a sensor. In the multiple targets case, various target formations were simulated. They included in-line, parallel, and crossing-target formations. The simulation results indicated that the cooperation between nodes is essential and reasonable tracking performance can be achieved even though the nodes do not communicate at every scan.

Extensive Monte-Carlo simulations were also performed with three selected scenarios. The resulting performance evaluation curves show the sensitivity of performance measures to certain parameter values. It was shown that the false alarm rate, if within reasonable range, does not impact the tracking performance significantly. On the other hand, sensor resolution plays an important role in the overall performance. Better sensor resolution not only improves tracking accuracy, but also helps dramatically in data-to-track and track-to-track association.

## 7.2 SUGGESTIONS FOR FUTURE RESEARCH

Although we have developed a general approach to distributed multitarget tracking, and demonstrated that it could be adapted for acoustic sensors, we have not answered all the technical questions relevant to a DSN. The following are some suggested directions for future research:

- *Incorporation of other sensor types.* Acoustic sensors can only achieve a certain tracking performance due to their inherent limitations such as resolution. Even with multiple sensors cooperating, tracks may still be missed, as illustrated in some of the scenarios simulated in this research. Addition of other passive (such as optical) or active (such as radar) sensors will improve tracking performance. The algorithms developed in this research can be extended to handle DSN nodes with different sensor types and it will be valuable to determine what kind of performance gains can be obtained.

- *Introduction of sensor control.* Especially with other sensor types, sensor control can improve performance by directing the sensors to the most valuable targets or more advantageous viewing angles. In a DSN, one can envision one node cuing another sensor node to look at a region, thereby optimizing the use of the sensor resources. This will be a crucial step in endowing the DSN with more intelligence and coupling distributed situation assessment with some planning.
- *Further communication related studies.* Although the research modeled communication connectivity, other features of a real communication network such as communication errors, delays, etc., were not considered. It will be useful to study the effects of these on overall system performance, as well as what actions the nodes can perform to mitigate their effects.
- *Applications to other distributed problems.* Other potential candidates that may use distributed situation assessment systems include strategic defense, undersea and ocean surveillance, etc. The DSN work performed in this and earlier projects can provide the foundation for developing new algorithms or predicting overall system performance. In particular, in strategic defense, communication problems may preclude sensor nodes (or battle managers) from having the same data. This will require the nodes to resolve their inconsistent views through explicit coordination.

We believe these are worthwhile topics for future research that will greatly enhance our understanding about distributed sensor networks.

## 8. REFERENCES

- [1] C.Y. Chong, S. Mori, E. Tse, and R.P. Wishner, "Distributed Hypothesis Formation in Distributed Sensor Networks," AI&DS Final Report TR-1015-2, Advanced Information & Decision Systems, Mountain View, CA, May 1983.
- [2] S. Mori, C.Y. Chong, E. Tse, and R.P. Wishner, "Multi-Target Multi-Sensor Tracking Problems: Part I: A General Solution and a Unified View on Bayesian Approaches," AI&DS Technical Report TR-1048-01, Advanced Information & Decision Systems, Mountain View, Ca., June 1983.
- [3] C.Y. Chong, S. Mori, M.R. Fehling, K.C. Chang, D.S. Spain, and R.P. Wishner, "Distributed Hypothesis Testing in Distributed Sensor Networks," ADS Final Report TR-1048-03, Advanced Decision Systems, Mountain View, CA, January 1986.
- [4] D.B. Reid, "An Algorithm for Tracking Multiple Targets," , *IEEE Trans. on Automat. Contr.*, Vol. AC-24, No. 6, pp.843-854, Dec. 1979
- [5] C.Y. Chong, S. Mori, E. Tse, and R.P. Wishner, "Distributed Hypothesis Formation in Distributed Sensor Networks," AI&DS Interim Report TR-1015-1, Advanced Information & Decision Systems, Mountain View, CA, December 1982.
- [6] S. Mori, C.Y. Chong, E. Tse, and R.P. Wishner, "Tracking and Classifying Multiple Targets without *A Priori* Identification," *IEEE Trans. on Automat. Contr.*, Vol. AC-31, No. 5, pp 401-409, May 1986.
- [7] C.Y. Chong, S. Mori, and K.C. Chang, "Information Fusion in Distributed Sensor Networks," *Proc. of 1985 American Control Conference*, pp. 830-835, Boston, MA, June 1985.
- [8] R.R. Tenney and J.R. Delaney, "A Distributed Aeroacoustic Tracking Algorithm," *Proc. of the 1984 American Control Conference*, pp. 1440 - 1450, San Diego, CA., June 1984.

[9] R.T. Lacoss, *Distributed Sensor Network - Semiannual Technical Summary*, ESD-TR-84-002, M.I.T. Lincoln Laboratory, Lexington, MA., June 1984.

[10] J.R. Delaney and R.R. Tenney, "Broadcast Communication Policies for Distributed Aeroacoustic Tracking," *Proc. of the 8th MIT/ONR Workshop on C<sup>3</sup> Systems*, pp. 195 - 199, Cambridge, MA, Dec. 1985.

[11] L.C. Ng and Y. Bar-Shalom, "Modeling of Unresolved Measurements for Multitarget Tracking," *Proc. OCEANS '81 Conf.*, Boston, MA, Sept. 1981.

[12] K.C. Chang and Y. Bar-Shalom, "Joint Probabilistic Data Association for Multitarget Tracking with Possibly Unresolved Measurements and Maneuvers," *IEEE Trans. on Automat. Contr.*, Vol. AC-23, pp. 585-594, July 1984.

[13] S. Mori, K.C. Chang, and C.Y. Chong, "Tracking Aircraft by Acoustic Sensors - Multiple Hypothesis Approach Applied to Possibly Unresolved Measurements," *Proc. American Control Conference*, pp. 1099-1105, June 1987.

[14] C.Y. Chong, K.C. Chang, and S. Mori, "Tracking Multiple Air Targets with Distributed Acoustic Sensors," *Proc. American Control Conference*, pp. 1831-1836, Minneapolis, MN, June 1987.

[15] C.Y. Chong, K.C. Chang, S. Mori, and D.S. Spain, "Tracking Air Targets by a Distributed Network of Acoustic Sensors," *1987 Tri-Service Data Fusion Symposium*, June 9-11, 1987.

## APPENDIX A. MERGED MEASUREMENT LIKELIHOOD CALCULATION

The likelihood of a measurement  $y$  originating from two existing tracks,  $\bar{\tau}_1$  and  $\bar{\tau}_2$ , is the joint probability density of  $y$ , the event  $M$  of track merging and event  $D$  of target detection, and is expanded as

$$P(y, M, D | \bar{\tau}_1, \bar{\tau}_2) = \int P(y | M, D, x_1, x_2, \bar{\tau}_1, \bar{\tau}_2) P(M | D, x_1, x_2, \bar{\tau}_1, \bar{\tau}_2) \cdot P(D | x_1, x_2, \bar{\tau}_1, \bar{\tau}_2) P(x_1 | \tau_1) P(x_2 | \tau_2) dx_1 dx_2 \quad (\text{A.1})$$

When we identify  $D$  with the event in which  $s_M^i \stackrel{\Delta}{=} s_1 + w_s^i \geq s_{TH}$  for  $i=1$  and  $2$ , we have

$$P(D | x_1, x_2, \bar{\tau}_1, \bar{\tau}_2) = P(D | s_1, s_2) = \left[ 1 - \operatorname{erf}\left(\frac{s_{TH} - s_1}{\sigma_s}\right) \right] \left[ 1 - \operatorname{erf}\left(\frac{s_{TH} - s_2}{\sigma_s}\right) \right] \quad (\text{A.2})$$

The track merging event is written as  $M = \{ |\phi_M^1 - \phi_M^2| < \delta\phi \}$ , and hence we have

$$P(M | D, x_1, x_2, \bar{\tau}_1, \bar{\tau}_2) = \operatorname{erf}\left(\frac{\delta\phi - (\phi_1 - \phi_2)}{\sqrt{(\sigma_\phi^1)^2 + (\sigma_\phi^2)^2}}\right) - \operatorname{erf}\left(\frac{-\delta\phi - (\phi_1 - \phi_2)}{\sqrt{(\sigma_\phi^1)^2 + (\sigma_\phi^2)^2}}\right) \quad (\text{A.3})$$

where  $\sigma_\phi^i$  is the standard deviation determined by eqn (3.4) for each  $i$ . The first factor in the integrand in (A.1) is then written as

$$P(y | M, D, x_1, x_2, \bar{\tau}_1, \bar{\tau}_2) = P(\phi_M^m | M, \phi_1, \phi_2) P(s_M^m | D, s_1, s_2) \quad (\text{A.4})$$

where  $\phi_M^m$  and  $s_M^m$  are defined by eqns. (3.5) - (3.7). In the first factor of the right hand side of (A.4), the conditioning on  $D$  was dropped because  $\phi_M^m$  can be defined as being independent from the detection event  $D$ . Similarly, in the second factor, the conditioning on  $M$  was dropped because  $s_M^m$  can be considered to be defined by

(3.7) regardless of whether or not the actual merging occurs. When we approximate (3.7) by (4.7), we have

$$P(s_M^m | D, s_1, s_2, \bar{q}) = \frac{g\left(s_M^m - h_s^m(s_1, s_2; \bar{s}_1, \bar{s}_2); \sigma_s^m(\bar{s}_1, \bar{s}_2)\right)}{1 - \operatorname{erf}\left(\frac{s_{TH} - h_s^m(s_1, s_2; \bar{s}_1, \bar{s}_2)}{\sigma_s^m(\bar{s}_1, \bar{s}_2)}\right)} \quad (\text{A.5})$$

The denominator of the right hand side of (A.5) is necessary because the range for  $s_M^m$  is  $[s_{TH}, \infty)$ .

Furthermore, we may approximately equate the right hand side of (A.2) with the denominator of the right hand side of (A.5). Then, since the GTSD factor and the SPTSD factor of a track are independent from each other, we have

$$\begin{aligned} & P(y, M, D | \bar{\tau}_1, \bar{\tau}_2) \\ &= \int P(\phi_M^m | M, \phi_1, \phi_2) \left[ \operatorname{erf}\left(\frac{\delta\phi - (\phi_1 - \phi_2)}{\sqrt{(\sigma_\phi^1)^2 + (\sigma_\phi^2)^2}}\right) - \operatorname{erf}\left(\frac{-\delta\phi - (\phi_1 - \phi_2)}{\sqrt{(\sigma_\phi^1)^2 + (\sigma_\phi^2)^2}}\right) \right] \\ & \quad g\left(s_M^m - h_s^m(s_1, s_2; \bar{s}_1, \bar{s}_2); \sigma_s^m(\bar{s}_1, \bar{s}_2)\right) P(x_1 | \tau_1) P(x_2 | \tau_2) dx_1 dx_2 \\ &= \int P(\phi_M^m | M, \phi_1, \phi_2) \\ & \quad \left[ \operatorname{erf}\left(\frac{\delta\phi - (\phi_1 - \phi_2)}{\sqrt{(\sigma_\phi^1)^2 + (\sigma_\phi^2)^2}}\right) - \operatorname{erf}\left(\frac{-\delta\phi - (\phi_1 - \phi_2)}{\sqrt{(\sigma_\phi^1)^2 + (\sigma_\phi^2)^2}}\right) \right] P(\phi_1 | \bar{\tau}_1) P(\phi_2 | \bar{\tau}_2) d\phi_1 d\phi_2 \\ & \quad \int g\left(s_M^m - h_s^m(s_1, s_2; \bar{s}_1, \bar{s}_2); \sigma_s^m(\bar{s}_1, \bar{s}_2)\right) P(s_1 | \bar{\tau}_1) P(s_2 | \bar{\tau}_2) ds_1 ds_2 \quad (\text{A.6}) \end{aligned}$$

The last integral in (A.6) can be easily calculated and yields (4.13). On the other hand, since  $\phi_M^m$  and the track merging event  $M$  are correlated, the calculation of the first integral in the last expression of (A.6) is not so straightforward. But, according to [11, 12], we have

$$\begin{aligned}
& \int P(\phi_M^m | M, \phi_1, \phi_2) \left[ \operatorname{erf}\left(\frac{\delta\phi - (\phi_1 - \phi_2)}{\sqrt{(\sigma_\phi^1)^2 + (\sigma_\phi^2)^2}}\right) - \operatorname{erf}\left(\frac{-\delta\phi - (\phi_1 - \phi_2)}{\sqrt{(\sigma_\phi^1)^2 + (\sigma_\phi^2)^2}}\right) \right] \\
& \quad \cdot P(\phi_1 | \bar{\tau}_1) P(\phi_2 | \bar{\tau}_2) d\phi_1 d\phi_2 \\
& = g(\phi_M - h_\phi^m(\bar{\phi}_1, \bar{\phi}_2; \bar{q}); \tilde{\sigma}_\phi^m) \left[ \operatorname{erf}\left(\frac{\delta\phi - \tilde{\Delta}\phi}{\tilde{\sigma}_{\Delta\phi}}\right) - \operatorname{erf}\left(\frac{-\delta\phi - \tilde{\Delta}\phi}{\tilde{\sigma}_{\Delta\phi}}\right) \right] \quad (\text{A.7})
\end{aligned}$$

which yields (4.9).

## APPENDIX B. GEOMETRY IN 2-D SUBSONIC ACOUSTIC TRACKING

### 1. Cartesian Coordinate

The two-dimensional position and velocity are:

$$u = [u_N, u_E]^T \quad (\text{B.1})$$

$$v = [v_N, v_E]^T$$

where

$u_N$  : northing,

$u_E$  : easting,

$v_N$  : northern component of velocity, and

$v_E$  : eastern component of velocity.

### 2. Polar Coordinate

The system orientation is determined as

$$u_N = R \cos \theta \quad u_E = R \sin \theta \quad (\text{B.2})$$

$$v_N = V \cos \psi \quad v_E = V \sin \psi \quad (\text{B.3})$$

where

$R$  : Range,

$\theta$  : Azimuth (from the north clockwise),

$V$  : Speed (scalar velocity), and

$\psi$  : Heading (from the north clockwise).

### 3. Acoustic Time Delay

The delayed position is defined as

$$u^d(t) \triangleq u(t - \delta(t)) \approx u(t) - \delta(t)v(t) \quad (\text{B.4})$$

where  $c$  is the (constant) speed of sound and the sound propagation delay and  $\delta(t)$  is a solution  $\Delta t$  to the nonlinear equation

$$|u(t-\Delta t)| = c \Delta t \quad (\text{B.5})$$

which has a unique solution if  $|\dot{u}(t)| = |\dot{v}(t)| < c$  (subsonic) for all  $t$ .  
The symbol  $|\cdot|$  is the norm on Euclidean spaces.

#### 4. Variables and Relations

The (delayed) acoustic range is

$$r(t) = R(t - \delta(t)) \quad (\text{B.6})$$

and the (delayed) acoustic azimuth is

$$\phi(t) = \theta(t - \delta(t)) \quad (\text{B.7})$$

The important derived quantities are:

- $\xi \triangleq V/c$  : Mach number,
- $\alpha \triangleq \theta - \psi$  : Aspect angle, and
- $\beta \triangleq \phi - \psi$  : Acoustic (delayed) aspect angle.

Assuming a constant velocity, eqn. (B.5) implies

$$R = Mr \quad (\text{B.8})$$

$$\sin(\phi - \theta) = \xi \sin \alpha \quad (\text{B.9})$$

where

$$M = \sqrt{1 + \xi^2 + 2\xi \cos \beta} \quad (\text{B.10})$$

The other useful variables are

$$N = \xi \cos \alpha \quad (\text{B.11})$$

and

$$D = \cos(\phi - \theta) = \sqrt{1 - (\xi \sin \alpha)^2} = M^{-1} (1 + \xi \cos \beta) \quad (\text{B.12})$$

Useful relations are:

$$M = N + D \quad (\text{B.13})$$

$$\sin\beta = M \sin\alpha \quad (\text{B.14})$$

$$D^2 - N^2 = 1 - \xi^2 \quad (\text{B.15})$$

$$\delta = \frac{r}{c} = \frac{R}{Mc} \quad (\text{B.16})$$

We should note that  $|\phi - \theta| \leq \pi/2$ , and therefore,  $\cos(\phi - \theta) \geq 0$ , and that  $1 - \xi \leq M \leq 1 + \xi$ , and  $N \leq D \leq 1$ .

### 5. Global Track Initiation Equations

Let us index two sensors by  $i \in \{1, 2\}$  and let the sensor  $i$ 's position be  $u_i^s = [u_{iN}^s, u_{iE}^s]^T$ . All the local variables are subscripted by  $i$ . It then follows from

$$u = u_i^s + r_i \left( \begin{bmatrix} \cos\phi_i \\ \sin\phi_i \end{bmatrix} + c^{-1} \begin{bmatrix} v_N \\ v_E \end{bmatrix} \right) \quad (\text{B.17})$$

and

$$\dot{\phi} = - \frac{V}{r_i} \frac{\sin\beta_i}{1 + \xi \cos\beta_i} = - \frac{V}{R_i} \frac{\sin(\phi_i - \psi_i)}{\cos(\phi_i - \theta_i)} \quad (\text{B.18})$$

that

$$F(\Phi)x = g(\Phi) - c^{-1} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} f(x) \quad (\text{B.19})$$

where  $x \triangleq [u, v]^T$ ,  $\Phi \triangleq [\phi_1, \phi_2, \dot{\phi}_1, \dot{\phi}_2]^T$ ,

$$F(\Phi) \triangleq \begin{bmatrix} \sin\phi_1 & -\cos\phi_1 & c^{-1}u_{1E}^s & -c^{-1}u_{1N}^s \\ \sin\phi_2 & -\cos\phi_2 & c^{-1}u_{2E}^s & -c^{-1}u_{2N}^s \\ \dot{\phi}_1 \cos\phi_1 & \dot{\phi}_1 \sin\phi_1 & \sin\phi_1 & -\cos\phi_1 \\ \dot{\phi}_2 \cos\phi_2 & \dot{\phi}_2 \sin\phi_2 & \sin\phi_2 & -\cos\phi_2 \end{bmatrix} \quad (\text{B.20})$$

$$g(\Phi) \triangleq \begin{bmatrix} \sin\phi_1 u_{1N}^s - \cos\phi_1 u_{1E}^s \\ \sin\phi_2 u_{2N}^s - \cos\phi_2 u_{2E}^s \\ \dot{\phi}_1(\cos\phi_1 u_{1N}^s + \sin\phi_1 u_{1E}^s) \\ \dot{\phi}_2(\cos\phi_2 u_{2N}^s + \sin\phi_2 u_{2E}^s) \end{bmatrix} \quad (\text{B.21})$$

and

$$f(x) \triangleq u_N v_E - u_E v_N \quad (\text{B.22})$$

The equ. (B.19) is reduced to a one-dimensional quadratic equation.

### 6. Partial Derivatives (1)

The transformation from the cartesian to the polar coordinates implies

$$\frac{\partial(R\theta)}{\partial u} = \begin{bmatrix} \cos\theta & \sin\theta \\ -R^{-1}\sin\theta & R^{-1}\cos\theta \end{bmatrix} \quad (\text{B.23})$$

and

$$\frac{\partial(V\psi)}{\partial v} = \begin{bmatrix} \cos\psi & \sin\psi \\ -V^{-1}\sin\psi & V^{-1}\cos\psi \end{bmatrix} \quad (\text{B.24})$$

with the Jacobians being

$$\det\left(\frac{\partial(R\theta)}{\partial u}\right) = R^{-1} \quad \text{and} \quad \det\left(\frac{\partial(V\psi)}{\partial v}\right) = V^{-1} \quad (\text{B.25})$$

### 7. Partial Derivatives (2)

The first derivatives of  $(r, \phi)$  are

$$\frac{\partial r}{\partial(R, \theta)} \stackrel{z}{=} [M^{-1}, \frac{r\xi}{D}\sin\alpha] \stackrel{z}{=} [M^{-1}, -\frac{(u_N v_E - u_E v_N)}{MDc}] \quad (\text{B.26})$$

$$\frac{\partial \mathbf{r}}{\partial (V, \psi)} = -\frac{r}{MDc} [\cos \beta, V \sin \beta] = -\frac{1}{MDc} \left[ \frac{u_N^d v_N + u_E^d v_E}{V}, u_E^d v_N - u_N^d v_E \right] \quad (\text{B.27})$$

$$\frac{\partial \phi}{\partial (R, \theta)} = \left[ 0, \frac{M}{D} \right] \quad (\text{B.28})$$

$$\frac{\partial \phi}{\partial (V, \psi)} = \frac{1}{Dc} [\sin \alpha, -V \cos \alpha] = \frac{1}{Dc} \left[ \frac{u_E v_N - u_N v_E}{RV}, \frac{u_N v_N + u_E v_E}{R} \right] \quad (\text{B.29})$$

$$\frac{\partial \mathbf{r}}{\partial \mathbf{u}} = \frac{1}{MD} [\cos \phi, -\sin \phi] = \frac{1}{MDr} [u_N^d, -u_E^d] \quad (\text{B.30})$$

$$\frac{\partial \mathbf{r}}{\partial \mathbf{v}} = -\frac{\delta}{MD} [\cos \phi, -\sin \phi] = -\frac{1}{MDc} [u_N^d, -u_E^d] \quad (\text{B.31})$$

$$\frac{\partial \phi}{\partial \mathbf{u}} = -\frac{M}{DR} [\sin \theta, -\cos \theta] = -\frac{M}{DR^2} [u_E, -u_N] \quad (\text{B.32})$$

$$\frac{\partial \phi}{\partial \mathbf{v}} = \frac{1}{Dc} [\sin \theta, -\cos \theta] = \frac{1}{DcR} [u_E, -u_N] \quad (\text{B.33})$$

with the Jacobians being

$$\det \left( \frac{\partial (\mathbf{r}, \phi, V, \psi)}{\partial (R, \theta, V, \psi)} \right) = \det \left( \frac{\partial (\mathbf{r}, \phi)}{\partial (R, \theta)} \right) = D^{-1} \quad (\text{B.34})$$

$$\det \left( \frac{\partial (\mathbf{r}, \phi, \mathbf{v})}{\partial (\mathbf{u}, \mathbf{v})} \right) = \det \left( \frac{\partial (\mathbf{r}, \phi)}{\partial \mathbf{u}} \right) = \frac{1}{DR} \quad (\text{B.35})$$

and

$$\det \left( \frac{\partial (\mathbf{r}, \phi, V, \psi)}{\partial (\mathbf{u}, \mathbf{v})} \right) = \frac{1}{DRV} \quad (\text{B.36})$$

### 8. Partial Derivatives (3)

The first derivatives of  $\dot{\phi}$  with respect to the  $(r, \phi, V, \psi)$  coordinate system are

$$\begin{aligned} \frac{\partial \dot{\phi}}{\partial(r, \phi, V, \psi)} &= \frac{1}{MD\delta} \left[ \frac{\xi \sin \beta}{r}, -\frac{N}{D}, -\frac{\xi \sin \beta}{MDV}, \frac{N}{D} \right] \\ &= \frac{V}{D^2 R^2} [M^2 D \sin \alpha, -R \cos \alpha, -\frac{R}{V} \sin \alpha, R \cos \alpha] \\ &= \frac{1}{DR} \left[ \frac{M^2 (u_E v_N - u_N v_E)}{R^2}, -\frac{Nc}{D}, -\frac{(u_E v_N - u_N v_E)}{DRV}, \frac{Nc}{D} \right] \end{aligned} \quad (B.37)$$

$$\begin{aligned} \frac{\partial \dot{\phi}}{\partial u} &= \frac{MV}{D^3 R^2} [(D \sin \alpha \cos \phi + \cos \alpha \sin \theta), -(D \sin \alpha \sin \phi + \cos \alpha \cos \theta)] \\ &= \frac{M}{D^3 R^4} [DM(u_E v_N - u_N v_E) u_N^d + (u_N v_N + u_E v_E) u_E, -(D u_E v_N - u_N v_E) u_N] \end{aligned} \quad (B.38)$$

and

$$\begin{aligned} \frac{\partial \dot{\phi}}{\partial u} &= -\frac{1}{D^3 R} [D \xi \sin \alpha \cos \phi + M \sin \theta, -(D \xi \sin \alpha \sin \phi + M \cos \theta)] \\ &= -\frac{1}{MD^3 R} \left[ D \xi \frac{u_E v_N - u_N v_E}{RV} \frac{u_N^d}{R} + \frac{u_E}{R}, -D \xi \frac{u_E v_N - u_N v_E}{RV} \frac{u_E^d}{R} - \frac{u_N}{R} \right] \end{aligned} \quad (B.39)$$

The Jacobians are

$$\det \left( \frac{\partial(\phi, \dot{\phi}, V, \psi)}{\partial(r, \phi, V, \psi)} \right) = \frac{\partial(\phi, \dot{\phi})}{\partial(r, \phi)} = -\frac{\partial \dot{\phi}}{\partial r} = -\frac{M^2 V}{D^3 R^2} \sin \alpha \quad (B.40)$$

$$\det \left( \frac{\partial(\phi, \dot{\phi}, V, \psi)}{\partial(R, \theta, V, \psi)} \right) = \det \left( \frac{\partial(\phi, \dot{\phi})}{\partial(r, \phi)} \right) \det \left( \frac{\partial(r, \phi)}{\partial(R, \theta)} \right) = -\frac{M^2 V}{D^4 R^2} \sin \alpha \quad (B.41)$$

and

$$\det\left(\frac{\partial(\dot{\phi}, \dot{\phi}, V, \psi)}{\partial(\mathbf{u}, \mathbf{v})}\right) = \det\left(\frac{\partial(\dot{\phi}, \dot{\phi})}{\partial \mathbf{u}}\right) \det\left(\frac{\partial(V, \psi)}{\partial \mathbf{v}}\right) = -\frac{M^2}{D^4 R^3} \sin \alpha \quad (\text{B.42})$$

APPENDIX C. TRACKING FROM COMMUNICATION AND  
INFORMATION THEORETIC VIEW POINT

***QUALCOMM, Inc.***

---

10555 Sorrento Valley Road  
San Diego, California 92121  
(619) 587-1121

# **Distributed Sensor Program**

**Final Report**

**by:**

**Jack K. Wolf**

**Andrew J. Viterbi**

**Greg P. Heinzinger**

**Audrey M. Viterbi**

**Glenn S. Dixon**

**December 19, 1986**

**Submitted To:**

**Advanced Decision Systems**

**201 San Antonio Circle, Suite 286**

**Mountain View, California 94040-1289**

# TABLE OF CONTENTS

1. INTRODUCTION .....	1-1
2. A RECURSIVE ESTIMATOR FOR QUANTIZED INPUTS .....	2-1
2.1 Introduction .....	2-1
2.2 Formulation of Linear Estimation Problem .....	2-1
2.3 Formulation of Quantized Estimation Problem .....	2-5
2.4 Exact Analysis of Quantized Estimator .....	2-9
2.5 Results .....	2-10
3. CONVENTIONAL ESTIMATION TECHNIQUES FOR .....	3-1
MULTI-TARGET/MULTI-SENSOR TRACKING	
4. A NEW APPROACH TO MULTI-TARGET/MULTI-SENSOR .....	4-1
TRACKING	
4.1 Problem Formulation .....	4-1
4.2 General Solution for Multiple Targets and One Sensor .....	4-2
4.3 Calculation of Branch Metrics .....	4-12
4.4 Further Considerations and Fusion of .....	4-14
Data From Two Sensors	
4.5 Examples .....	4-18
4.6 Conclusions .....	4-24
APPENDIX A - THE K BEST PATHS THROUGH A TRELLIS .....	A-1
APPENDIX B - TASK STATEMENTS FOR DISTRIBUTED .....	B-1
SENSOR PROGRAM	
REFERENCES	

## TABLE OF TABLES

Table 4.1: Statistics From Computer Simulation . . . . .	4-22
Table 4.2: Data For Example . . . . .	4-30
Table 4.2: Data For Example (continued) . . . . .	4-31
Table 4.2: Data For Example (continued) . . . . .	4-32
Table 4.2: Data For Example (continued) . . . . .	4-33
Table 4.3: Best Tracks . . . . .	4-34
Table A.1: State Metrics for Trellis of Figure A.1 . . . . .	A-3
Table A.2: Branch Metric Sum of Each Configuration . . . . .	A-5
Table A.3: State Metrics for Expanded Trellis of Figure A.8 . . . . .	A-10

## Section 1 Introduction

The problem of accurate determination of the tracking parameters of multiple targets using multiple distributed sensors is relatively recent in origin but already has an extensive bibliography [References 1 - 7]. Most of this appears in publications on control and estimation theories and reflects the methodology of these disciplines. In this study we have attempted to apply instead the techniques of the communication and information theorist to arrive at new approaches to apply to the solution of this very complex and demanding problem area.

The report reflects our various attempts at formulating the problem from a communication theorist's point of view. Three approaches are described and explored in Sections 2, 3, and 4. The order is chronological and each section is essentially self contained. Section 2 deals with the problem from the viewpoint of the communicator who is provided with a channel or channels to disseminate tracking data among various sensors, for which the channel capacity is much smaller than that required to fully characterize each track. In the simplified and idealized model considered, each sensor is allowed to transmit one bit about each sensor measurement to all other sensors. The degradation in the variance of the ultimate estimate based on all single bit quantized measurements is compared to that of a network which can distribute all measurements without quantization (but with measurement and channel noise). Results are moderately promising, particularly when the a priori measurement variance is large compared to the noise variance.

To proceed beyond this simplistic model it was necessary to review the literature and develop metrics for performance optimization and evaluation. For this purpose, the work described in Section 3 was undertaken. A number of simulations were performed of conventional estimation for the multi-target/multi-sensor problem based on a somewhat simplified model. While we were able to demonstrate that reasonable results could be achieved for two sensors tracking a single target with quantized data transfer between the two, no major conclusions were reached although the model and experience was useful for establishing the

## Section 2

### A Recursive Estimator for Quantized Inputs

#### 2.1 Introduction

The classical multi-sensor linear estimation problem assumes that a sequence of noisy measurements is made on a parameter or set of parameters, such that the measurements are linear transformations of the parameter set plus noise. The least squares estimator can be expressed as a recursive (Kalman filter) estimator based on each successive measurement, given the estimator for all previous measurements.

In an application where the sensors are widely separated and interconnected by a communication network with low capacity links, it may not be possible to provide the fusion point with all measurement samples, particularly if high accuracy (multiple bits/sample) is required. We consider a case where the network can exchange one bit of information among all users for every sample or estimate made by any user. We show that the conditional mean estimator for this limited network-capacity model satisfies a recursive relation very similar to the linear Kalman filter and we evaluate the effect of quantization on the ultimate accuracy and speed of convergence of the resulting estimate.

#### 2.2 Formulation of Linear Estimation Problem

We consider initially the scalar case of a single parameter  $S$  which is a priori Gaussian with zero mean and variance  $N_0$ . Suppose  $n$  measurements  $r_1, r_2, \dots, r_n$  are made with variances  $N_1, N_2, \dots, N_n$ . It is readily shown that the conditional mean (least squares) estimator based on all  $n$  measurements is

$$\hat{S} = \sum_{k=1}^n \alpha_k r_k \quad \text{where } \alpha_k = \frac{1/N_k}{\sum_{j=0}^n (1/N_j)} \quad (1)$$

and that its variance (mean square error) is

$$\hat{S}_1 = E(S|r_1) = M_1 = \frac{N_0}{N_0 + N_1} r_1 \quad (7)$$

and the resulting variance

$$E[(S - \hat{S}_1)^2 | r_1] = \text{Var}(S | r_1) \equiv V_1 = \frac{\int (S - M)^2 p(r_1/S) p(S) dS}{\int p(r_1/S) p(S) dS} \quad (8)$$

$$= \frac{N_0 N_1}{N_0 + N_1} = \frac{1}{1/N_0 + 1/N_1}$$

Continuing in this way, suppose we add a second measurement  $r_2$  and consider the new problem of estimating  $S - \hat{S}_1$  given  $r_2$ . Letting  $S^{(1)} = S - \hat{S}_1$ , we have

$$\hat{S}_2 = E(S^{(1)} | r_2) = \frac{\int S' p_2(r_2/S') p_{S^{(1)}}(S') dS'}{\int p_2(r_2/S') p_{S^{(1)}}(S') dS'} \quad (9)$$

Now since  $\hat{S}_1$  is a linear function of the Gaussian variable  $r_1$  and  $S$  is a priori Gaussian,  $S^{(1)}$  is also Gaussian with zero mean, since  $E(S^{(1)}) = E(S | r_1) - \hat{S}_1 = 0$  and variance  $V_1 = 1 / [1/N_0 + 1/N_1]$  since  $\text{Var}(S^{(1)}) = E[(S - \hat{S}_1)^2 | r_1] = \text{Var}(S | r_1) = V_1$

Now let  $r_2' = r_2 - \hat{S}_1 = (S - \hat{S}_1) + n_2 = S' + n_2$ .

Then

$$V_k = \frac{V_{k-1}}{1 + V_{k-1}/N_k} = V_{k-1} \left( 1 - \frac{1}{1 + N_k/V_{k-1}} \right) \quad (17)$$

The estimator can be implemented recursively accordingly to the block diagram of Figure 2.1.

This result is the simplest special case of the recursive Kalman filter approach to least squares estimation and could also be derived by the innovation sequence approach.

### 2.3 Formulation of Quantized Estimation Problem

Suppose now that instead of transmitting an analog quantity, each sensor can only transmit one bit (or at most a few bits) about each measurement. As assumed, the a priori mean of the variable  $S$  is zero. Hence let the first observer simply transmit the sign of the first measurement  $\text{sgn}(r_1)$  to all other observers. Each observer then computes the conditional mean of  $S$  given  $\text{sgn}(r_1)$

$$\begin{aligned} E(S | \text{sgn}(r_1)) &= \int_{-\infty}^{\infty} S p(S | \text{sgn}(r_1)) dS \\ &= \frac{\int_{-\infty}^{\infty} S P(\text{sgn}(r_1) | S) p(S) dS}{\int_{-\infty}^{\infty} P(\text{sgn}(r_1) | S) p(S) dS} \\ &= \frac{\int_{r_1 \geq 0} \int_{-\infty}^{\infty} S p(r_1 | S) p(S) dS}{\int_{r_1 \geq 0} \int_{-\infty}^{\infty} p(r_1 | S) p(S) dS} \end{aligned} \quad (18)$$

$$\begin{aligned}
E(S^2 | \text{sgn}(r_1)) &= \frac{\int \int_{r_1 \geq 0} S^2 p(r_1|S) p(S) dS}{\int \int_{r_1 \geq 0} p(r_1|S) p(S) dS} = 2 \int_{r_1 \geq 0} \left( \frac{N_0 N_1}{N_0 + N_1} + M^2 \right) \frac{e^{-r_1^2 / 2(N_0 + N_1)}}{\sqrt{2\pi(N_0 + N_1)}} dr_1 \\
&= \frac{N_0 N_1}{N_0 + N_1} + \left( \frac{N_0}{N_0 + N_1} \right)^2 \int_{-\infty}^{\infty} \frac{r_1^2 e^{-r_1^2 / 2(N_0 + N_1)}}{\sqrt{2\pi(N_0 + N_1)}} dr_1 = \frac{N_0 N_1}{N_0 + N_1} + \frac{N_0^2}{N_0 + N_1}
\end{aligned}$$

Hence from (20) and (21) it follows that

$$V_1 = \text{Var}(S | \text{sgn}(r_1)) = \frac{N_0 N_1}{N_0 + N_1} + \frac{N_0^2}{N_0 + N_1} \left( 1 - \frac{2}{\pi} \right) \quad (22)$$

Comparing this with the linear case (8) we find that the relative reduction in variance is only  $2/\pi$  times that in the linear case.

Now consider the second observer. He observes  $r_2$  and receives the one bit  $\text{sgn}(r_1)$ . Suppose he performs the estimate based on  $\text{sgn}(r_1)$ , just computed,

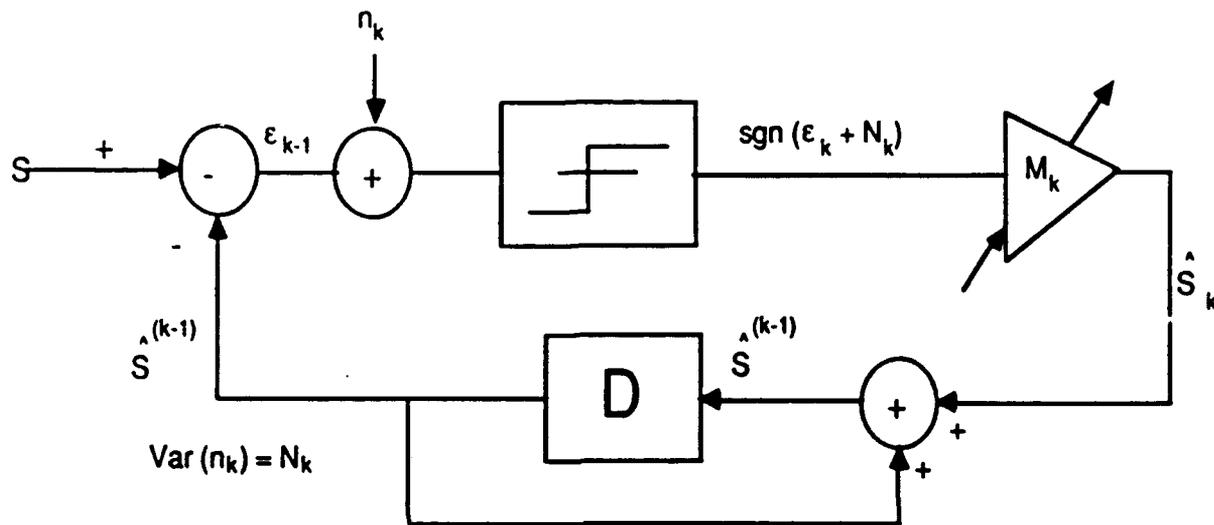
$$\hat{S}_1 = M_1 \text{sgn}(r_1)$$

$$\text{where } M_1 = \sqrt{\frac{2}{\pi} \frac{N_0}{1 + N_1/N_0}}$$

and upon observing  $r_2$  forms the new variable

$$r_2^{(1)} = r_2 - \hat{S}_1 = S + n_2 - \hat{S}_1 = S - E(S | \text{sgn}(r_1)) + n_2$$

Finally, suppose the second observer quantizes  $r_2^{(1)}$  to one bit. Then  $\text{sgn}(r_2^{(1)}) = \text{sgn}(r_2 - S_1)$  and sends this on to all other observers.



$$M_k = \sqrt{\frac{2}{\pi} \frac{V_{k-1}}{1 + N_k/V_{k-1}}}$$

$$V_k = V_{k-1} \left( 1 - \frac{2\pi}{1 + N_k/V_{k-1}} \right)$$

Figure 2.2: Quantized Recursive Estimator

variance source = 1.0  
variance noise = 0.1

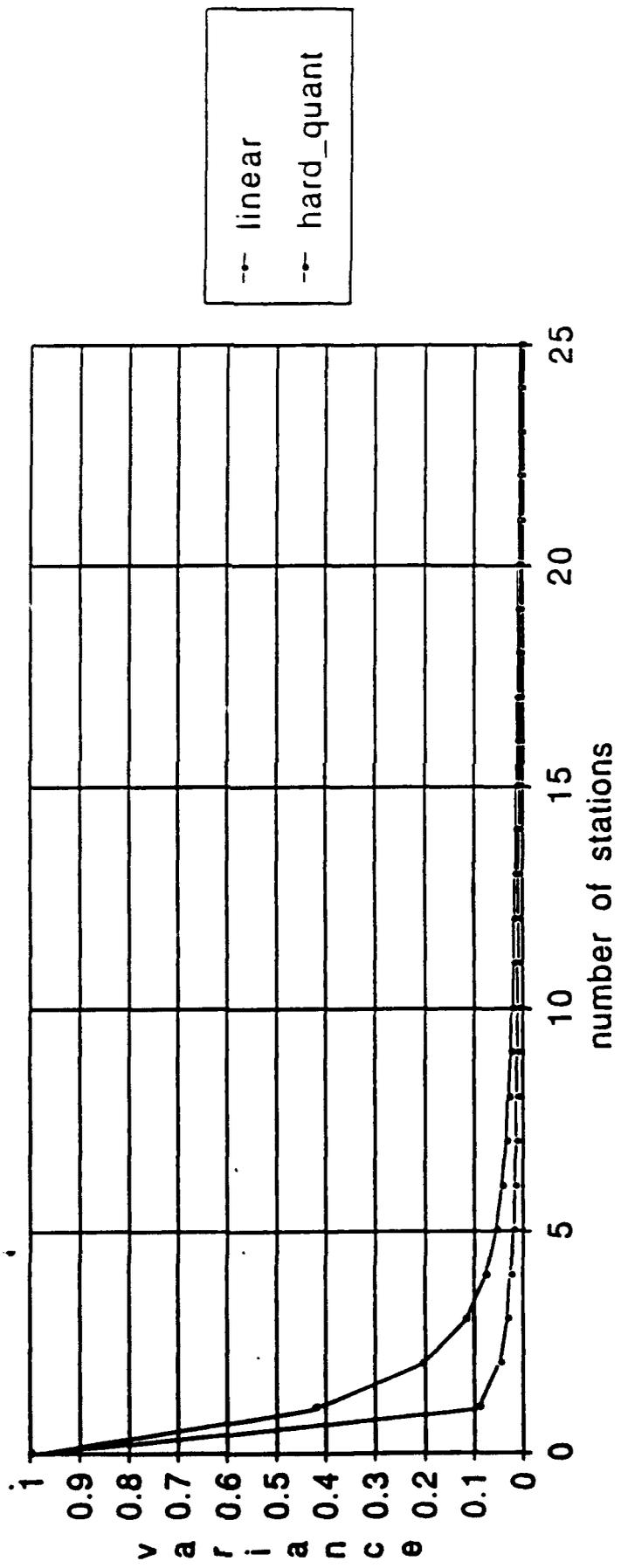


Figure 2.4: Measurement Error Variance for  $N_k = .1$

variance source = 1.0  
variance noise = 0.5

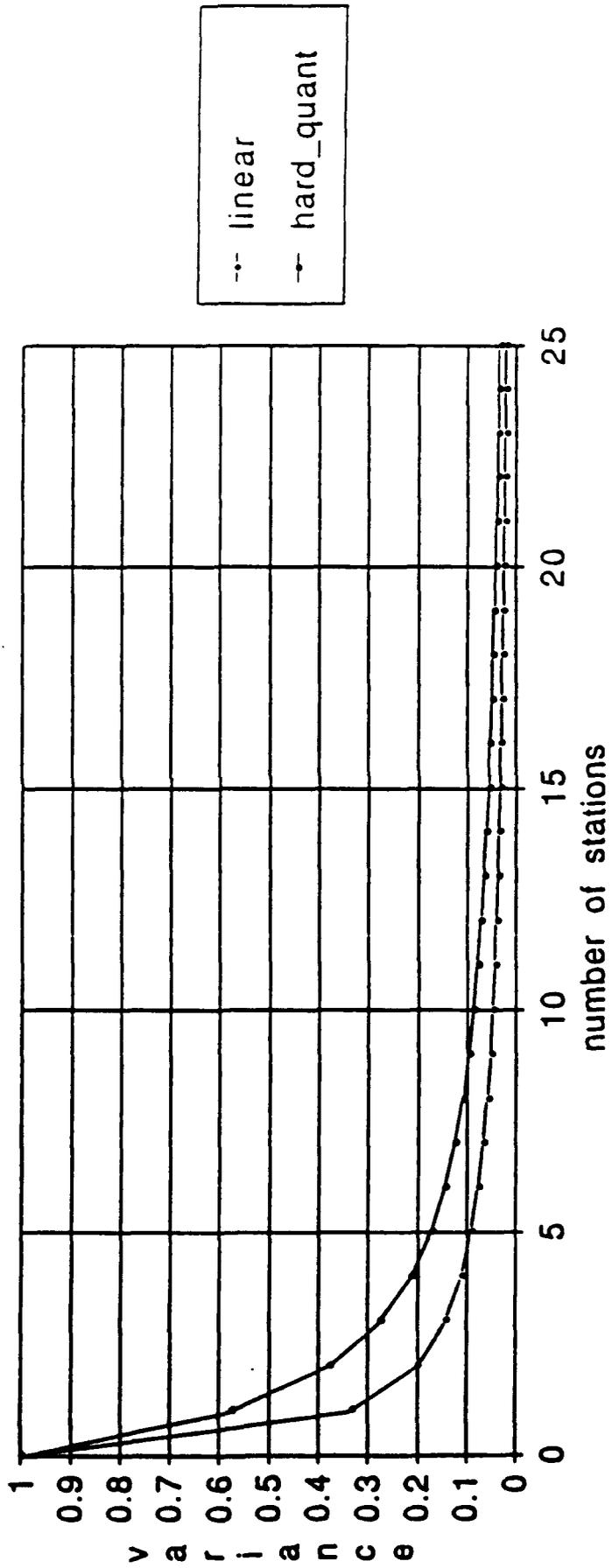


Figure 2.6: Measurement Error Variance for  $N_k = .5$

variance source = 1.0  
variance noise = 2.0

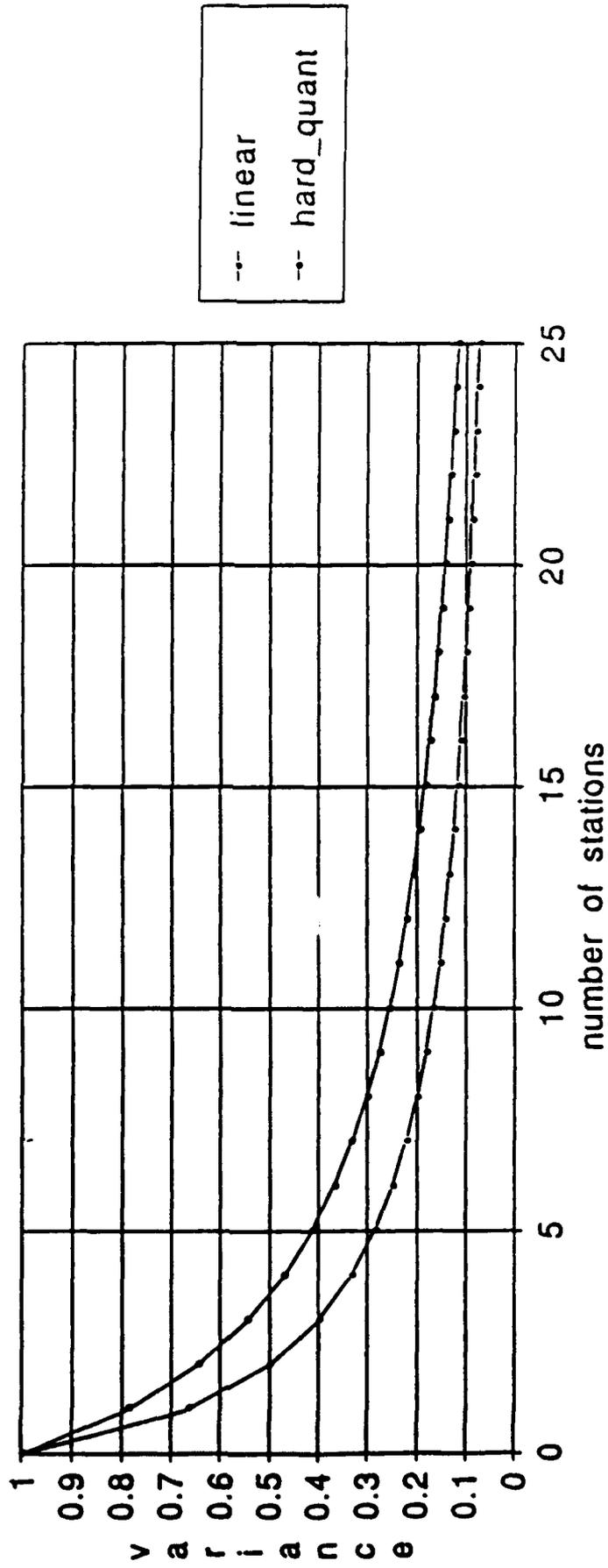


Figure 2.8: Measurement Error Variance for  $N_k = 2.0$

variance source = 1.0  
variance noise = 10.0

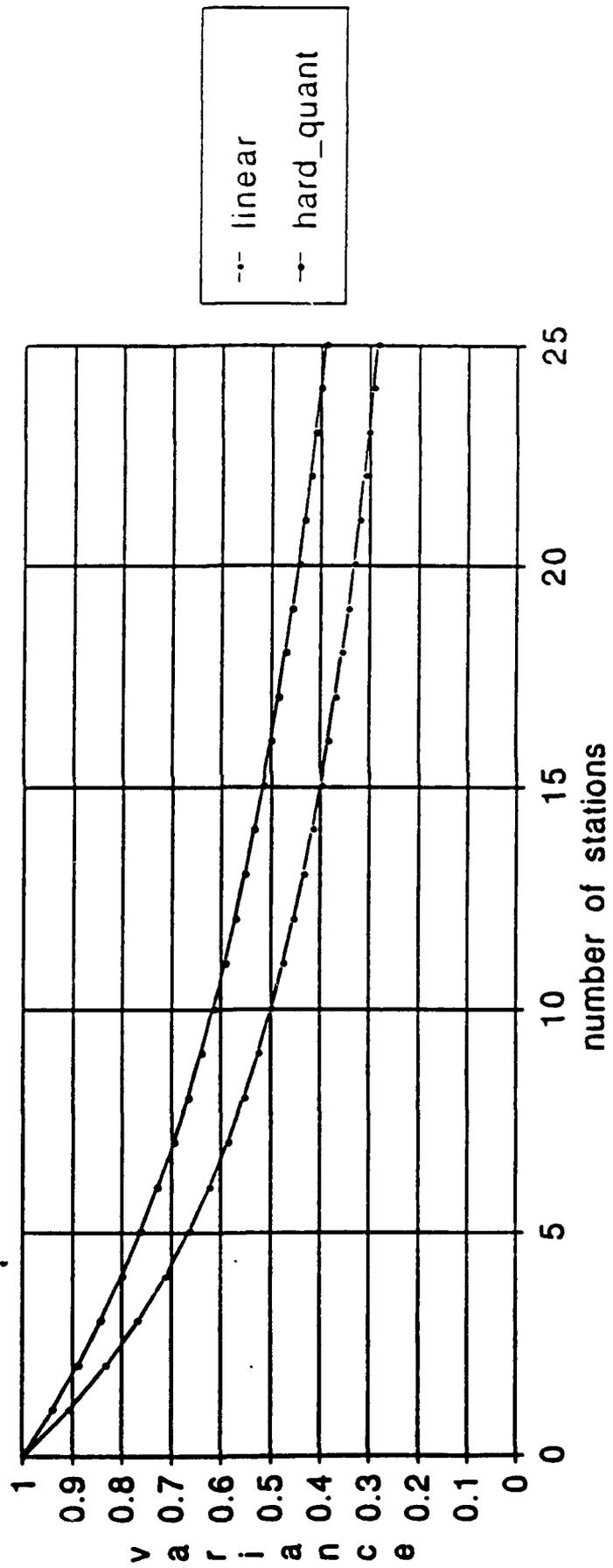


Figure 2.10: Measurement Error Variance for  $N_k = 10.0$

variance source = 1.0  
variance noise = 0.25

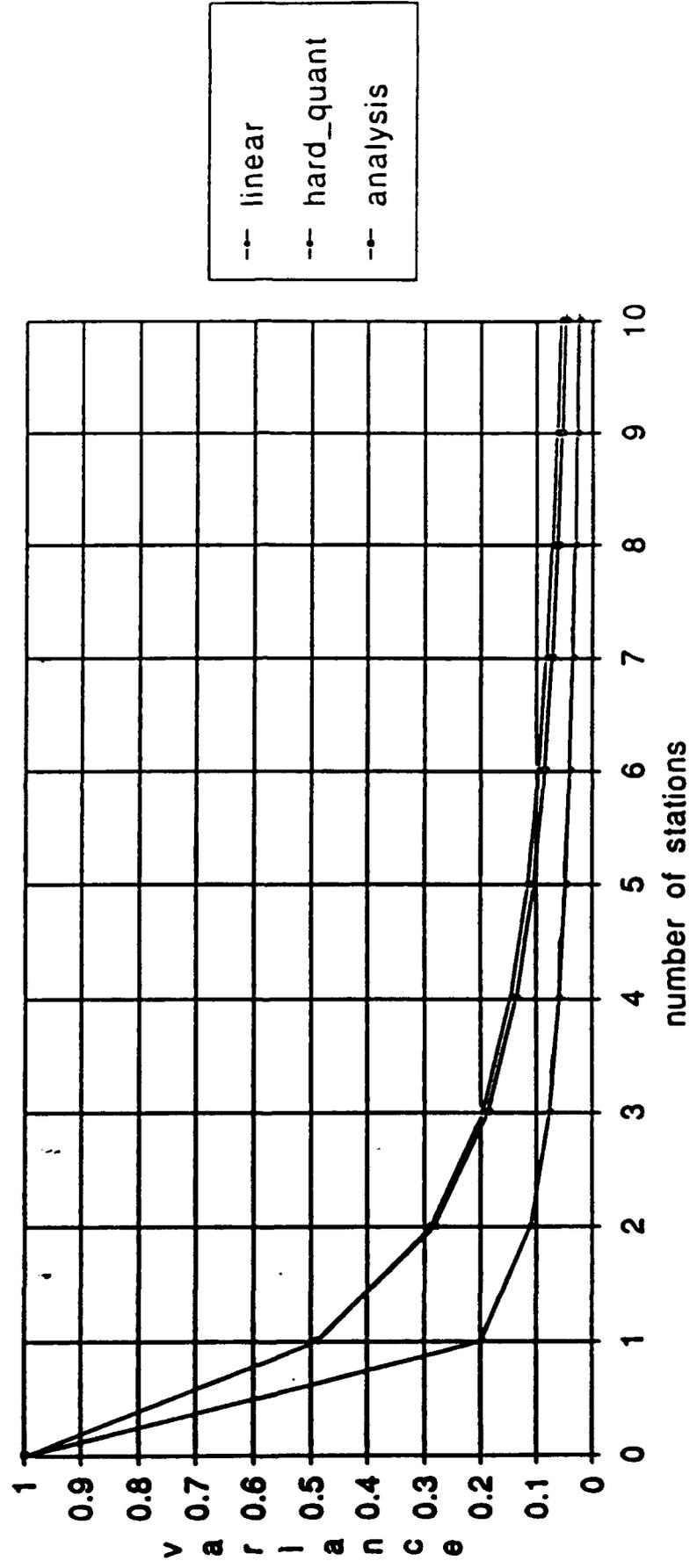


Figure 2.12: Comparison of Appropriate and Exact Analysis  $N_k = .25$

overlapping fields of view. This configuration was useful in simulating a fusion problem where the Kalman algorithm used data from the two sensors.

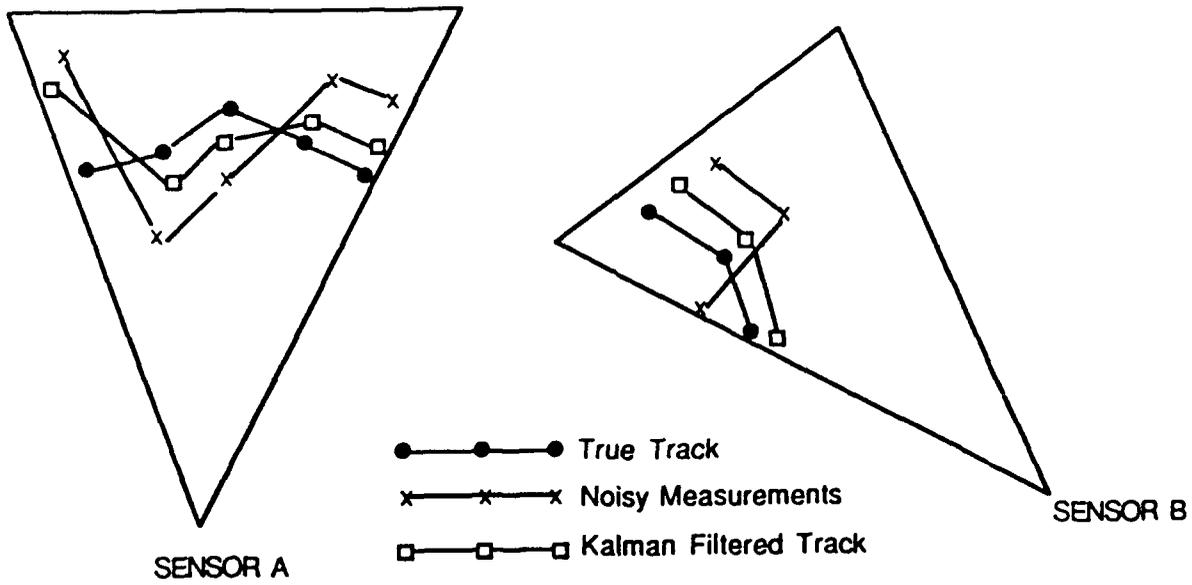
For each run of the simulation, three different versions of the track were displayed on the screen. These were:

- (a) Track associated with system model (equation (1)). This will be called the true track.
- (b) Track associated with measurements (equations (2) and (3)). This is the true track plus measurement noise.
- (c) Track produced by Kalman filtering the measurements.

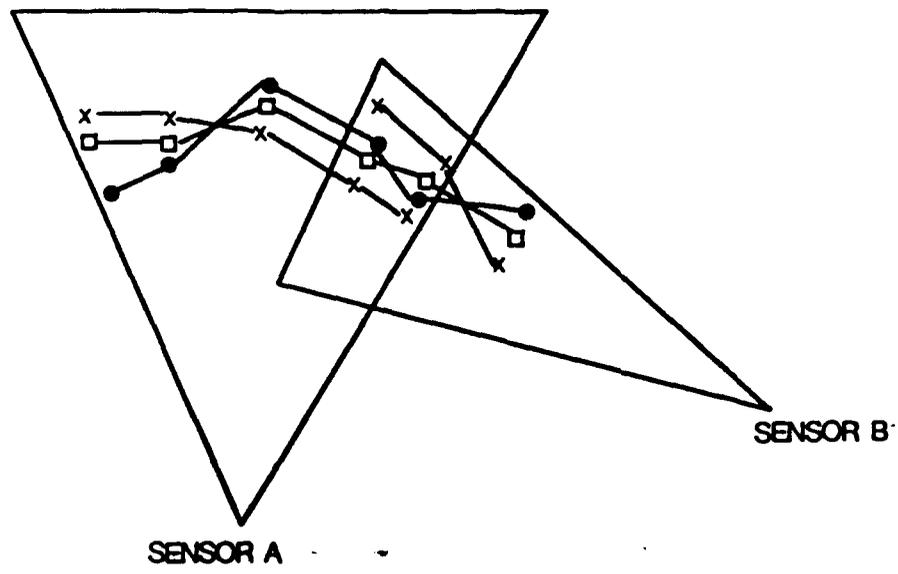
A sketch of these three tracks for two typical runs of the simulation are shown in Figure 3.1. Figure 3.1(a) gives a sketch of an output for the hand off configuration and Figure 3.1(b) depicts an output for the case of fusion. The simulation allowed for the introduction of clutter points and also allowed for missing measurements (i.e., a probability of detection strictly less than 1).

The effects of limiting the amount of information transmission between sensor A and sensor B for the hand off configuration were investigated. Here a track was produced (as in Figure 3.1(a)) which traveled through the field of view of sensor A and then at some later time entered the field of view of sensor B. If there were no limit to the amount of information that could be transmitted from sensor A to sensor B, sensor A would transmit all information required in the Kalman filtering algorithm to sensor B. With a limit to this information transmission, the questions investigated were what information should be sent and how to allocate the amount of information to the parameters which were chosen for transmission. Two schemes were investigated. In the first, the noisy measurements themselves were quantized and transmitted from sensor A to sensor B. The second scheme involved sensor A running a Kalman algorithm on its measurements and then transmitting to sensor B a quantized final state vector and error covariance matrix.

Only a very limited number of runs were attempted. It should be noted that the simulation of this Section was concerned only with the case of a single target. A



(a) Hand-Off Configuration



(b) Fusion Configuration

Figure 3.1: Typical Options From Simulation

## Section 4

### A New Approach to Multi-Target/Multi-Sensor Tracking

#### 4.1 Problem Formulation

Multi-target/multi-sensor tracking is concerned with detecting and estimating the tracks of an (unknown) number of targets using signals received by a fixed number of sensors. The problem as to how to process the signals received by the various sensors so as to obtain a best estimate of these tracks is very complex and not well understood. A number of papers have appeared [References 1- 7] giving partial solutions to this problem.

Here we consider a new approach to multi-target/multi-sensor tracking. To describe this approach we first consider multi-targets sensed by a single sensor. Later we will extend our approach to the case of two or more sensors.

We assume that the sensor measures the position and velocity of the various targets at a discrete set of time instants  $t_0, t_1, \dots, t_N$ . We will allow for false measurements (that is, measurements that do not correspond to targets) and missed measurements. We assume that the only measurements to be considered are paired measurements: that is, those for which we have information on both position and velocity. We assume that  $M_i$  paired measurements are made at time instant  $t_i$  and we denote these paired measurements as

$$(\vec{x}_{1,i}, \vec{v}_{1,i}), (\vec{x}_{2,i}, \vec{v}_{2,i}), \dots, (\vec{x}_{M_i,i}, \vec{v}_{M_i,i})$$

The number of these paired measurements which correspond to actual targets can be less than, equal to, or greater than  $M_i$ .

$\vec{x}_1$  and has velocity  $\vec{v}_1$  at time  $t_1$  and if a target progresses to position  $\vec{x}_2$  and has velocity  $\vec{v}_2$  at time  $t_2$ , then the energy (or cost) expended by the target during the time interval  $t_1 \leq t \leq t_2$ , denoted  $E(t_1, \vec{x}_1, \vec{v}_1; \vec{x}_2, \vec{v}_2)$ , can be expressed as a function of the two paired measurements at times  $t_1$  and  $t_2$ . Furthermore, the energies (or costs) add along a track so that the total energy (or cost) of a track corresponding to the measurements  $(\vec{x}_0, \vec{v}_0), (\vec{x}_1, \vec{v}_1), \dots, (\vec{x}_N, \vec{v}_N)$  is

$$\sum_{i=1}^N E(t_{i-1}, \vec{x}_{i-1}, \vec{v}_{i-1}, t_i, \vec{x}_i, \vec{v}_i).$$

Our proposed solution to Task 2 is to connect the paired measurements

$$\begin{aligned} &(\vec{x}_{1,0}, \vec{v}_{1,0}), (\vec{x}_{2,0}, \vec{v}_{2,0}), \dots, (\vec{x}_{K,0}, \vec{v}_{K,0}), \\ &(\vec{x}_{1,1}, \vec{v}_{1,1}), (\vec{x}_{2,1}, \vec{v}_{2,1}), \dots, (\vec{x}_{K,1}, \vec{v}_{K,1}), \\ &\dots \qquad \dots \qquad \dots \qquad \dots \\ &(\vec{x}_{1,N}, \vec{v}_{1,N}), (\vec{x}_{2,N}, \vec{v}_{2,N}), \dots, (\vec{x}_{K,N}, \vec{v}_{K,N}) \end{aligned}$$

to form  $K$  tracks in such a way that the total cost or sum of the energies over the  $K$  tracks for the time interval  $t_0 \leq t \leq t_N$  is a minimum. Note that when we have created our  $K$  tracks, each paired measurement can be assigned to one and only one track and furthermore, each track must contain a paired measurement for each of the time instants  $t_0, t_1, \dots, t_N$ . The number of possible choices for such a set of  $K$  tracks is  $(K!)^N$  which grows as  $K^{KN}$  so that an exhaustive search for the  $K$  tracks with the minimum summed energy is not feasible even for moderate value of  $K$  and  $N$ . Here we demonstrate a computationally efficient algorithm for finding these best  $K$  tracks.

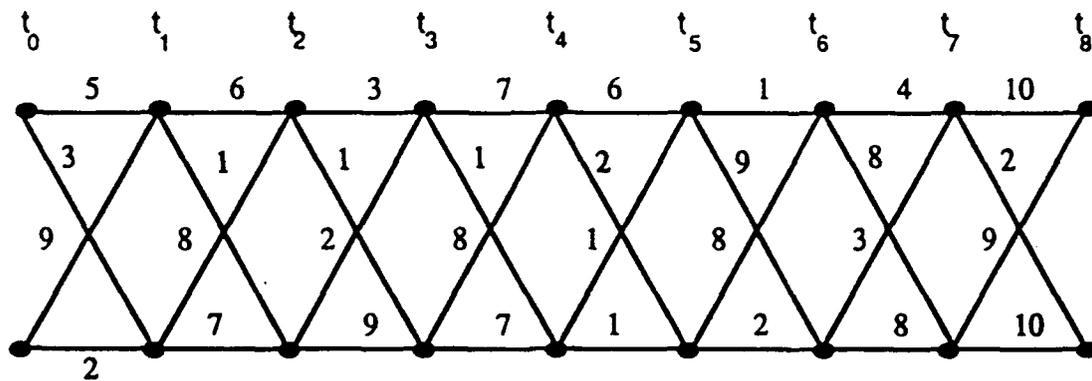


Figure 4.1: Trellis for  $K = 2$  and  $N = 8$

finding the two tracks which result in the minimum amount of expended energy is described in the Appendix. As a matter of fact, the choice of tracks given in Figure 4.2(b) is the best choice of two tracks (in the sense of the smallest total energy for the two targets).

Appendix A describes a simple method for finding the "best"  $K$  trajectories through a trellis for the more general situation where we have extraneous paired measurements (say due to clutter or other false targets). When there are no extraneous measurements i.e., when  $M_i = K$ , the algorithm is particularly simple. For each increment in time we choose the best (in the sense of minimum sum of the energies) of the  $K!$  different ways of connecting the states at depth  $(i - 1)$  to the states at depth  $i$ . Then the chosen branches at each depth are connected to form the best  $K$  trajectories. This procedure leads to the solution given in Figure 4.2(b) for the case of the trellis shown in Figure 1.

For the case of clutter, i.e., when  $M_i > K$  for some  $i$ , the situation is more complicated. Now we form an expanded trellis with

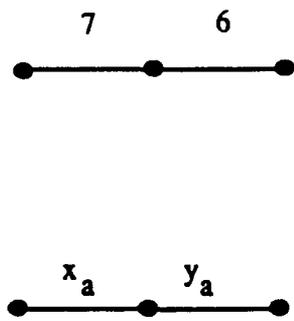
$$\binom{M_i}{K}$$

states at depth  $i$ ,  $i = 0, 1, 2, \dots, N$ . Each state at depth  $i$  in the expanded trellis corresponds to  $K$  of the  $M_i$  paired measurements at time  $t_i$ . For each of  $K$  paired measurements taken at time  $t_{i-1}$  and for each set of  $K$  paired measurements at time  $t_i$  ( $i = 1, 2, \dots, N$ ) the best of the  $K!$  different ways of connecting these paired measurements is determined and the sum of energies corresponding to this best connection is determined. The branch which connects the states of the expanded trellis corresponding to these two sets of  $K$  paired measurements is labeled by this sum. The Viterbi algorithm is then used to find the best path through the expanded trellis. The final result is that we find the  $K$  paths through the original trellis which has the smallest sum of energies. Appendix A contains the details of this procedure and some illustrative examples.

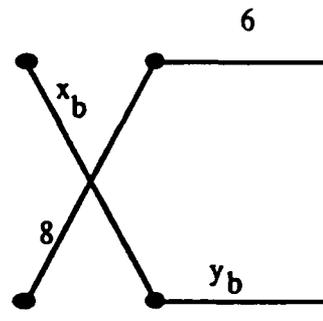
Although several approaches to the solution to this problem were considered, only one approach will be followed here. The approach requires that we know the number of missing states at each depth in the trellis. The general idea behind this approach is that the missing states are filled in with fictitious states chosen so as to provide the smallest minimum summed energy for the resulting  $K$  paths.

We illustrate this approach for the example shown in Figure 4.4. Since by assumption we know that there is a missing state of depth 4, we know there is a paired measurement missing at time  $t_4$ . Our approach is to create a best fictitious measurement where "best" here means that we choose the best paired measurement and the best pair of paths which minimize the total energy for both paths. This involves calculating the energy for all branches which enter and exit from the fictitious state. These energies are calculated as a function of the position and velocity of this fictitious measurement. For each possible manner by which the two paths can pass through this fictitious state, one has a best choice for the fictitious measurements. The four possibilities are shown in Figure 4.4, for the missing state shown in Figure 4.3. In each case the energy on the branch entering the fictitious state is labeled by the variable  $x$  and the energy on the branch leaving the fictitious state is labeled by the variable  $y$ . The sum of the energies for the four situations are: (a)  $13 + x_a + y_a$ ; (b)  $14 + x_b + y_b$ ; (c)  $8 + x_c + y_c$ ; (d)  $9 + x_d + y_d$ . For each of these four situations we can find the choice of the fictitious paired measurement that made the sum of the energies a minimum and then use that one that gives the minimum sum over all four choices.

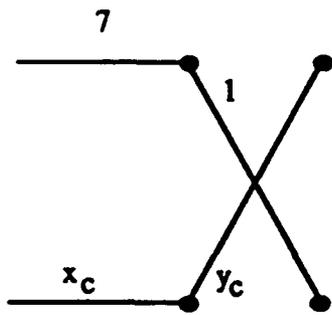
For the case of one missing measurement and  $K$  paths, one would have  $(K!)^2$  different possible paths to consider in choosing the best value for this fictitious measurement. The generalization of this technique for more than one missing measurement is straight forward in principle but very complex in practice.



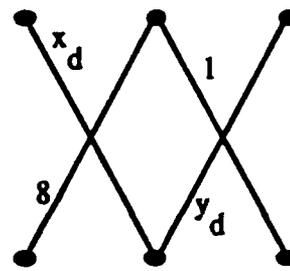
(a)



(b)



(c)



(d)

**Figure 4.4: Possible Paths Through Missing State**

measurements. Since the integral can be written as a sum of integrals, one in each dimension, we can solve the problem in one dimension and sum the results. Thus from here on, we can treat the acceleration as a scalar. Since the acceleration  $a(t)$  is the derivation of the velocity  $v(t)$  and since  $v(t)$  must satisfy the equation

$$\int_{t_1}^{t_2} v(t) dt = x(t_2) - x(t_1) \quad (2)$$

(where  $x(t)$  is the position of the target at time  $t$ ), by the use of Lagrange multipliers we can minimize the quantity

$$\Lambda_c = \int_{t_1}^{t_2} [v'(t)^2 + \lambda v(t)] dt \quad (3)$$

Using calculus of variations we find that  $v'(t)$  satisfies the equation.

$$\lambda - 2 \frac{d}{dt} (v'(t)) = 0 \quad (4)$$

This yields the solution,

$$a(t) = v'(t) = \frac{\lambda}{2} t + a \quad (5)$$

$$v(t) = \frac{\lambda}{4} t^2 + at + b \quad (6)$$

Here  $a$  and  $b$  are unknowns which are obtained using the boundary conditions  $v(t_1) = v_1$  and  $v(t_2) = v_2$ . This yields

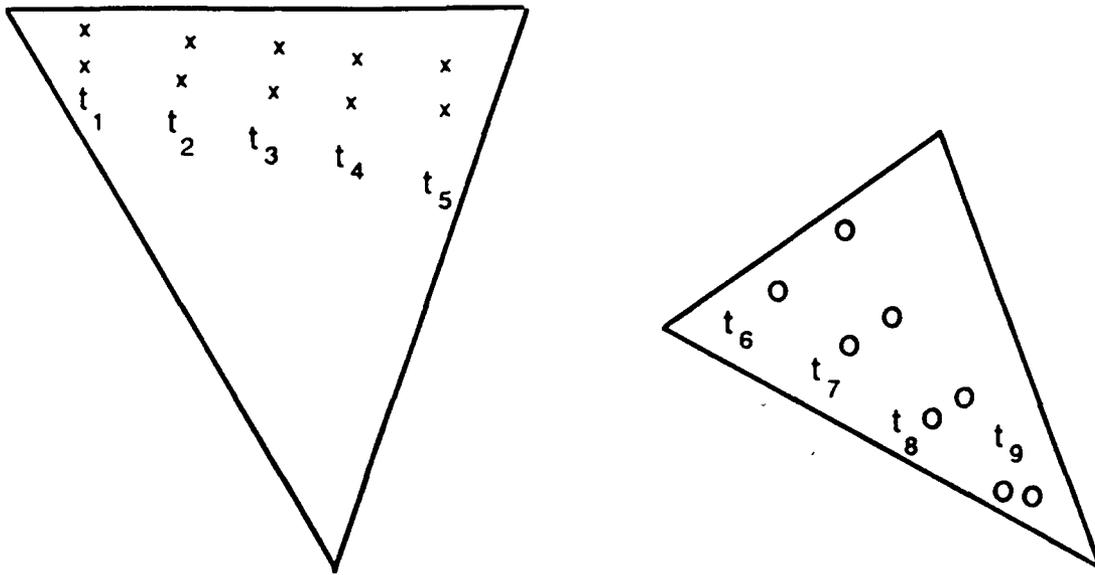
$$a = \frac{(v_2 - v_1) - \frac{\lambda}{4} (t_2^2 - t_1^2)}{t_2 - t_1} \quad b = \frac{v_1 t_2 - v_2 t_1 - \frac{\lambda}{4} t_1 t_2 (t_1 - t_2)}{t_2 - t_1} \quad (7)$$

The Lagrange multiplier  $\lambda$  can be found by substituting  $v(t)$  into the constraint equation (2). Simplifying,  $\lambda$  is given by,

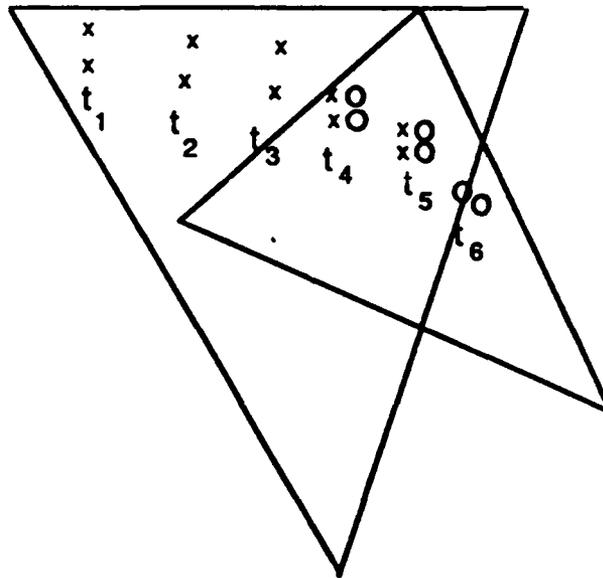
to each set. This pre-screening seems feasible since if one were to observe many targets on a radar screen, ambiguity with regard to tracks would occur only for small sets of these targets. (That is, two targets on opposite sides of the radar screen would not need to be separated.) No study was made as to how to accomplish this pre-screening in an automatic manner.

Still another issue relates to the fact that our measurements of position and velocity are based upon noisy observations and thus contain error terms. This suggests that we may be able to use some type of filtering and/or prediction to improve these measurements. However, we are faced with the problem that the filtering operation can only be applied to individual tracks and we are using the measurements to determine the tracks. In certain cases it is possible to alternate between the filtering problem and the track discrimination problem. One case where this can be done is where there is no clutter. Then the best  $K$  paths to depth  $i$  in the trellis can be determined based only upon the paired measurements taken to time  $t_i$ . The paired measurements at time  $t_i$  can then be modified based upon filtering the measurements corresponding to the  $K$  tracks to that time. Finally, the modified measurements at time  $t_i$  can be used in computing the energies corresponding to the branches which connect these states to the states at depth  $(i + 1)$  and the process can be iterated. The value of such a procedure needs to be determined by computer simulations. Again, time did not permit such an investigation.

We next consider various situations which arise when we have measurements from two or more sensors. Our discussion assumes exactly two sensors but our ideas easily generalize to the case of an arbitrary number of sensors. Two very different configurations are depicted in Figure 4.5. In Figure 4.5a, termed the "hand-off" configuration, we show two sensors with non-overlapping volume coverage. The left sensor produces paired measurements, denoted by  $x$ 's, at times  $t_1, t_2, t_3, t_4$  and  $t_5$  and the right sensor produces paired measurements, denoted by  $o$ 's, at times  $t_6, t_7, t_8$ , and  $t_9$ . ( $t_i > t_j$  for  $i > j$ ). Figure 4.5b depicts the "fusion" configuration whereby the volume coverages of the two sensors partially overlap. Here the left sensor produces measurements at times  $t_1, t_2, t_3, t_4$  and  $t_5$  while the right sensor produces measurements at times  $t_4, t_5$  and  $t_6$ . (Again,  $t_i > t_j$  for  $i > j$ ).

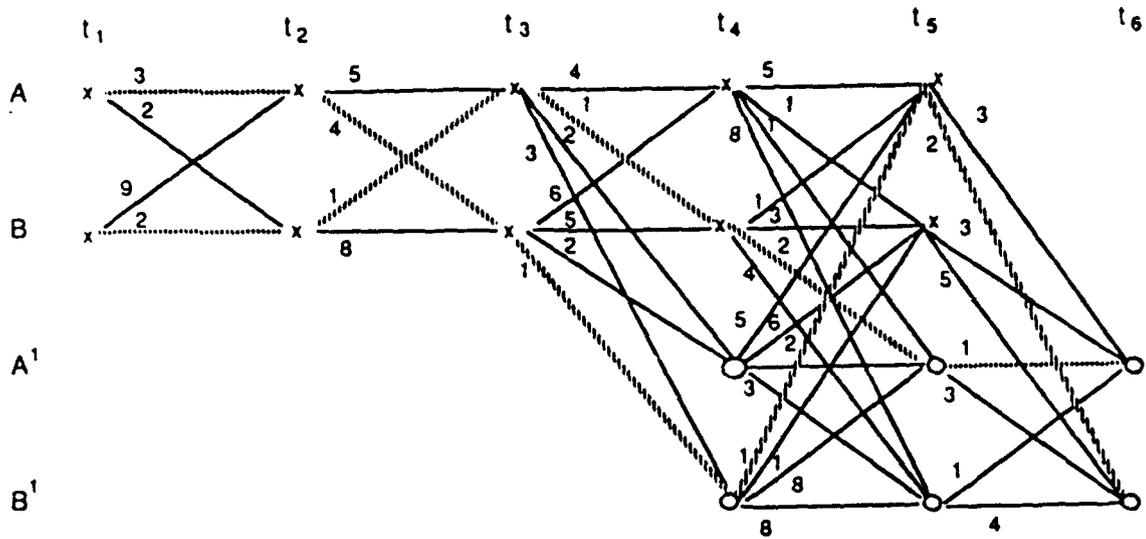


a) The "Hand-Off" Configuration

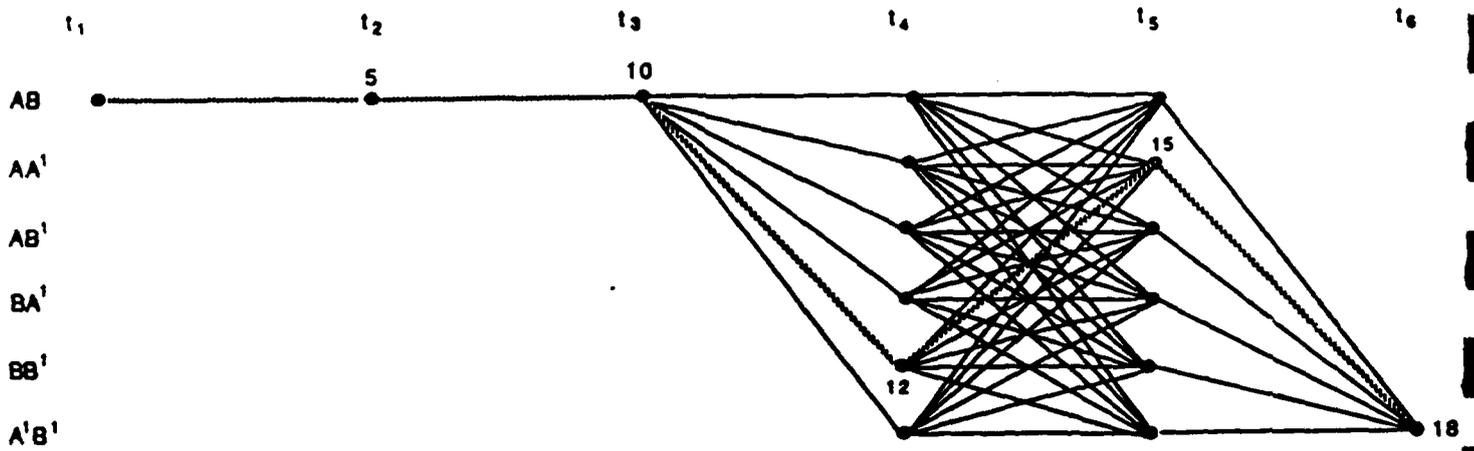


b) The "Fusion" Configuration

Figure 4.5: Two Configurations For Two Sensors



(a) Single Trellis Formed from Measurements from Two Sensors



(b) Expanded Trellis for Measurements from Two Sensors ( $K = 2$ )

Figure 4.6: Two Sensor Problem - Fusion

Type of Trajectory: Straight line or Parabola

Noise Variance: 0.5 or 2.0

Clutter Radius: 5.0 or 20.0 (both position and velocity)

As stated above, 100 runs were taken for each of these eight cases. A description of the results follows:

We first consider how this algorithm can be used to determine the number of targets when this number is not known. It should be remembered in every case, the correct number of targets was equal to two but that there were two extra clutter measurements at each time instant. Thus the algorithm must investigate the possibilities: number of targets = 1, 2, 3 or 4.

The results are given in Table 4.1 for the 8 cases considered. We describe the results in the first column of this table. Two parabolic tracks were considered. The position and velocity clutter radius was equal to 20 and the noise variance (per component) was equal to 0.5. When the best single path was found, its total cost (averaged over 100 runs) was equal to 598.7. When the two best paths were found, their average total costs (then averaged over 100 runs) was equal to 677.1 while the larger of the two costs (averaged over 100 runs) was equal to 756.1. When the three best paths were found, their average total cost (then averaged over 100 runs) was equal to 16,621.3 while the largest of the three costs (averaged over 100 runs) was equal to 26,750.4. This sharp jump in costs indicates that the number of tracks is equal to two. Thus, we see that in this case:

- (a) The largest of the costs is the more sensitive indicator of the correct number of tracks, and
- (b) There is a clear indication of the correct number of tracks.

By observing the seven other cases, we see that similar but less dramatic results are obtained. Thus it appears that we have obtained a viable solution finding the number of targets (i.e., Task 1).

The performance of the algorithm for a given run will now be discussed. We consider the situation where the clutter radius (position and velocity) is equal to 20, the noise variance (on each coordinate of the position and velocity) is equal to 2.0, and the two targets have intersecting parabolic tracks. Measurements were taken at twenty time instants. The  $(x, y)$  coordinates of the two targets at the twenty time instants are shown in Figure 4.7. At each time instant there were two noisy target measurements and two clutter measurements for a total of eighty measurements. The  $(x, y)$  coordinate of these eighty measurements are shown in Figure 4.8. The time of each measurement and the velocity associated with each measurement are not shown in this Figure. Figure 4.9 shows the best single track as found by the algorithm. The track consisted of all the noisy measurement for one of the targets (Target 2). The total cost of this track was 2,556.2. Figure 4.10 shows the best two tracks as found by the algorithm. Each of these tracks consisted of the noisy measurements for each of the targets. One track was the best single track as shown in Figure 4.9 and had a cost of 2,556.2. The other track had a cost of 3,070.6. Figure 4.11 depicts the best three tracks as found by the algorithm. Each of these tracks was made up of a mixture of target measurements and clutter points. The total cost of each of these three tracks was: 10,074.6; 25,010.5; and 26,823.9. The high cost of the worst track indicates that the true number of tracks is equal to 2 and that Figure 4.10 represents the best separation of these two tracks.

The data for this example is given in Table 4.2 and 4.3. Table 4.2 gives the position  $(x, y)$  and velocity  $(v_x, v_y)$  for target 1, target 2, clutter point 1 and clutter point 2 at each of the twenty time instants. Table 4.3 gives the best tracks as found by the algorithm when the number of tracks is 1, 2, 3 and 4. This table also gives the cost of these tracks.

A more dramatic example of the efficiency of this algorithm is shown in Figures 4.12 through 4.15. Here a clutter radius of 50 and a noise variance of 5 was used. Again, there were two true targets having intersecting parabolic tracks as depicted in Figure

# Target Trajectories

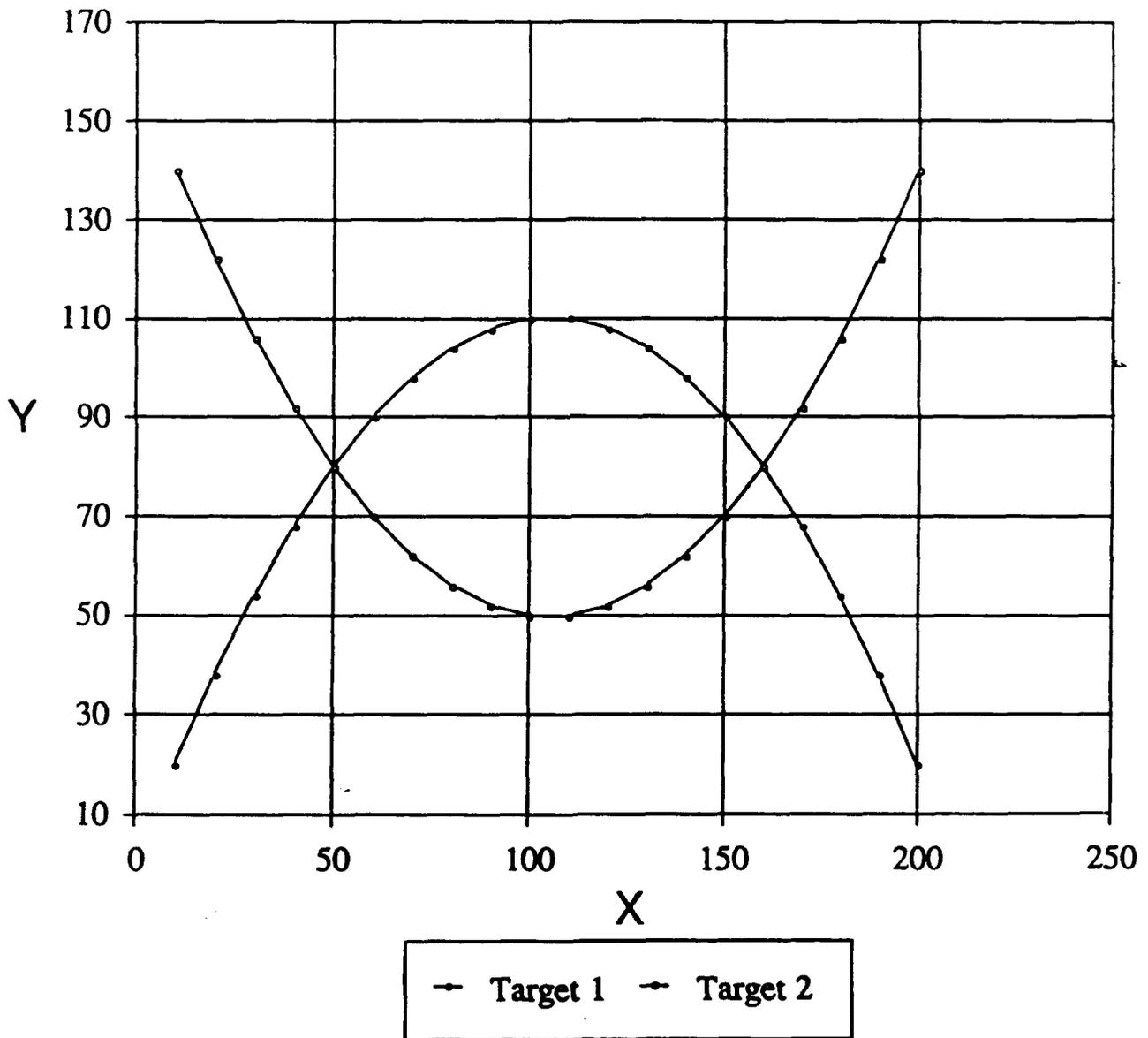


Figure 4.7: True Tracks

# Radar Measurements

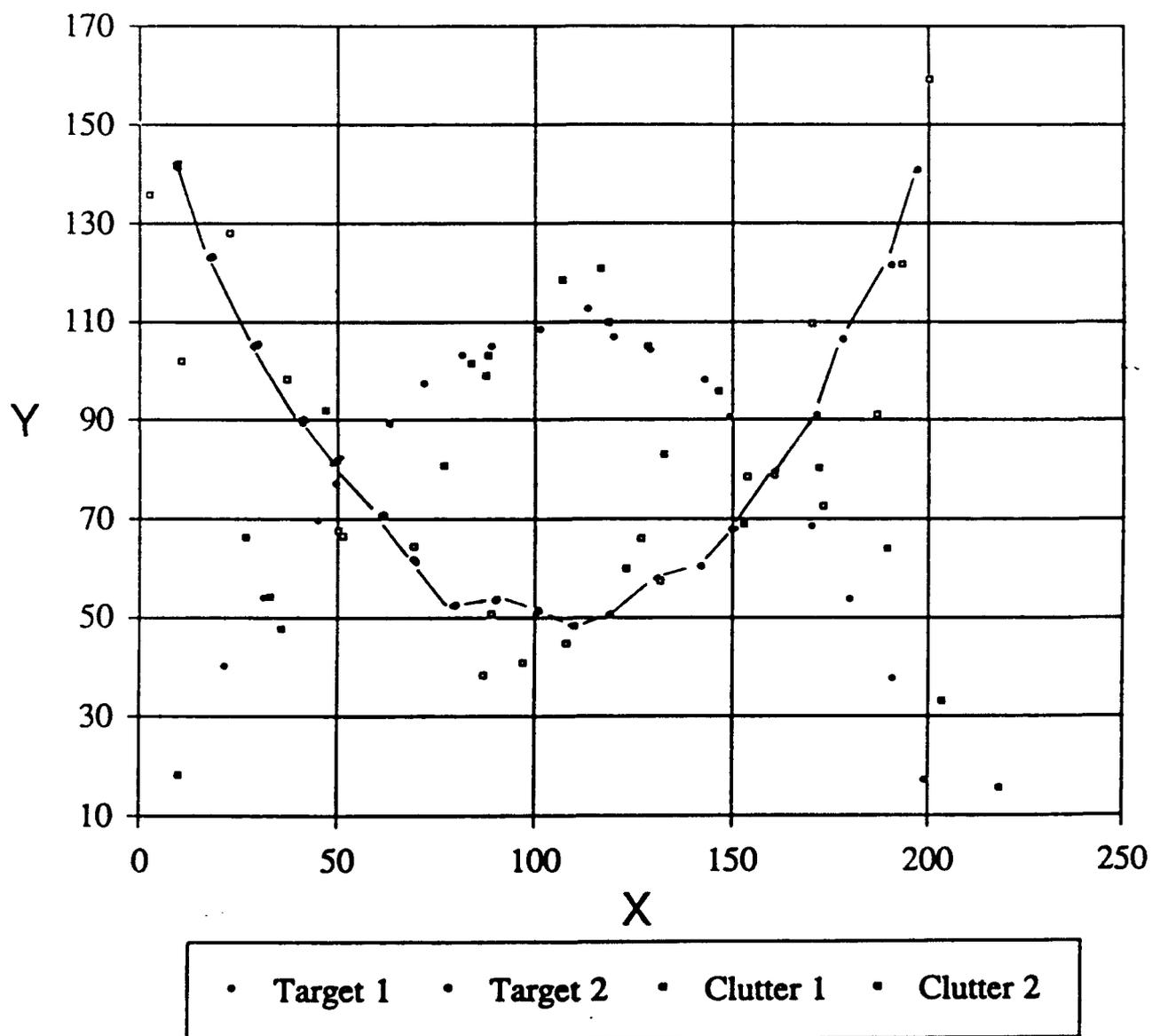


Figure 4.9: Best Single Track

# Radar Measurements

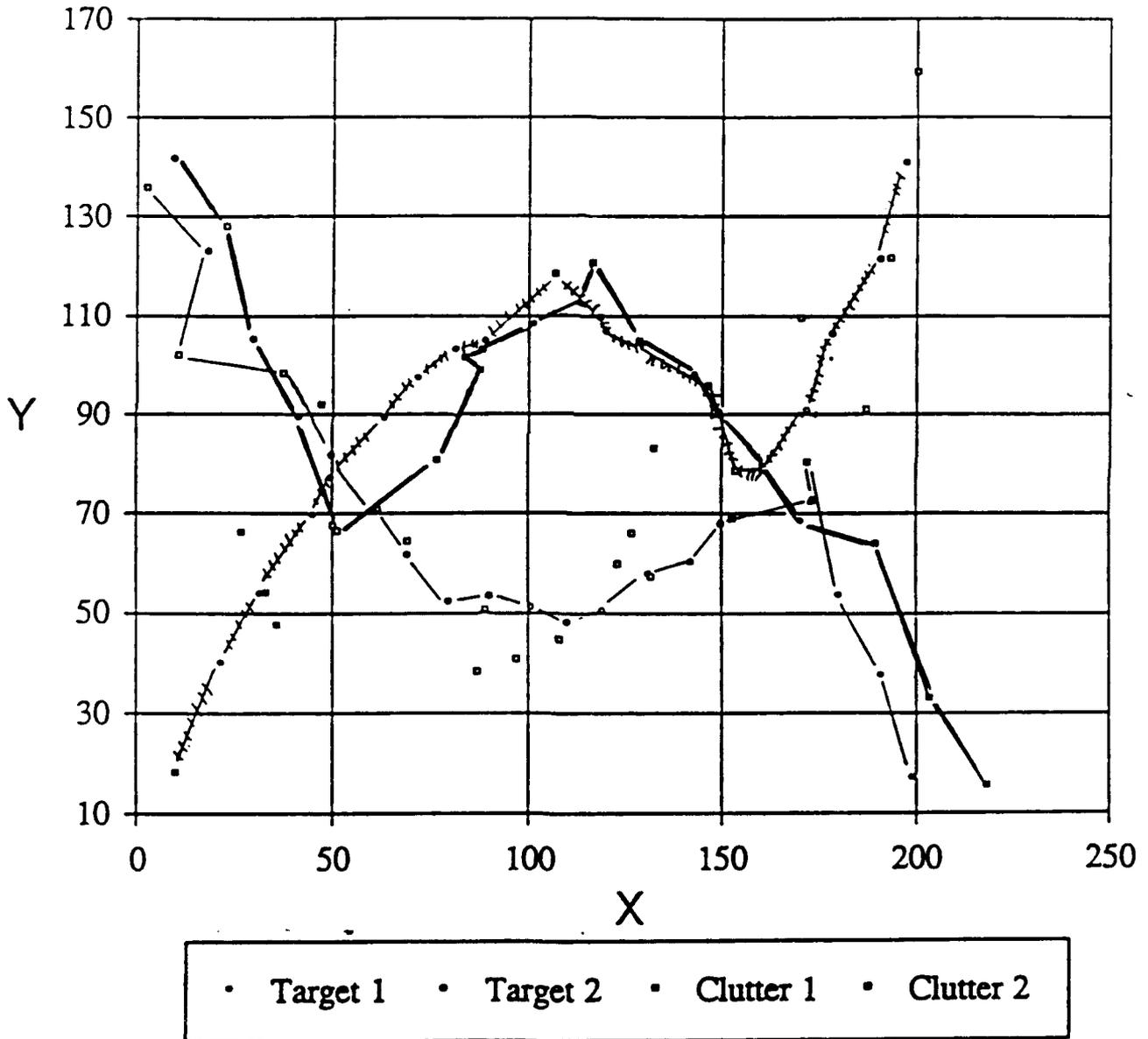


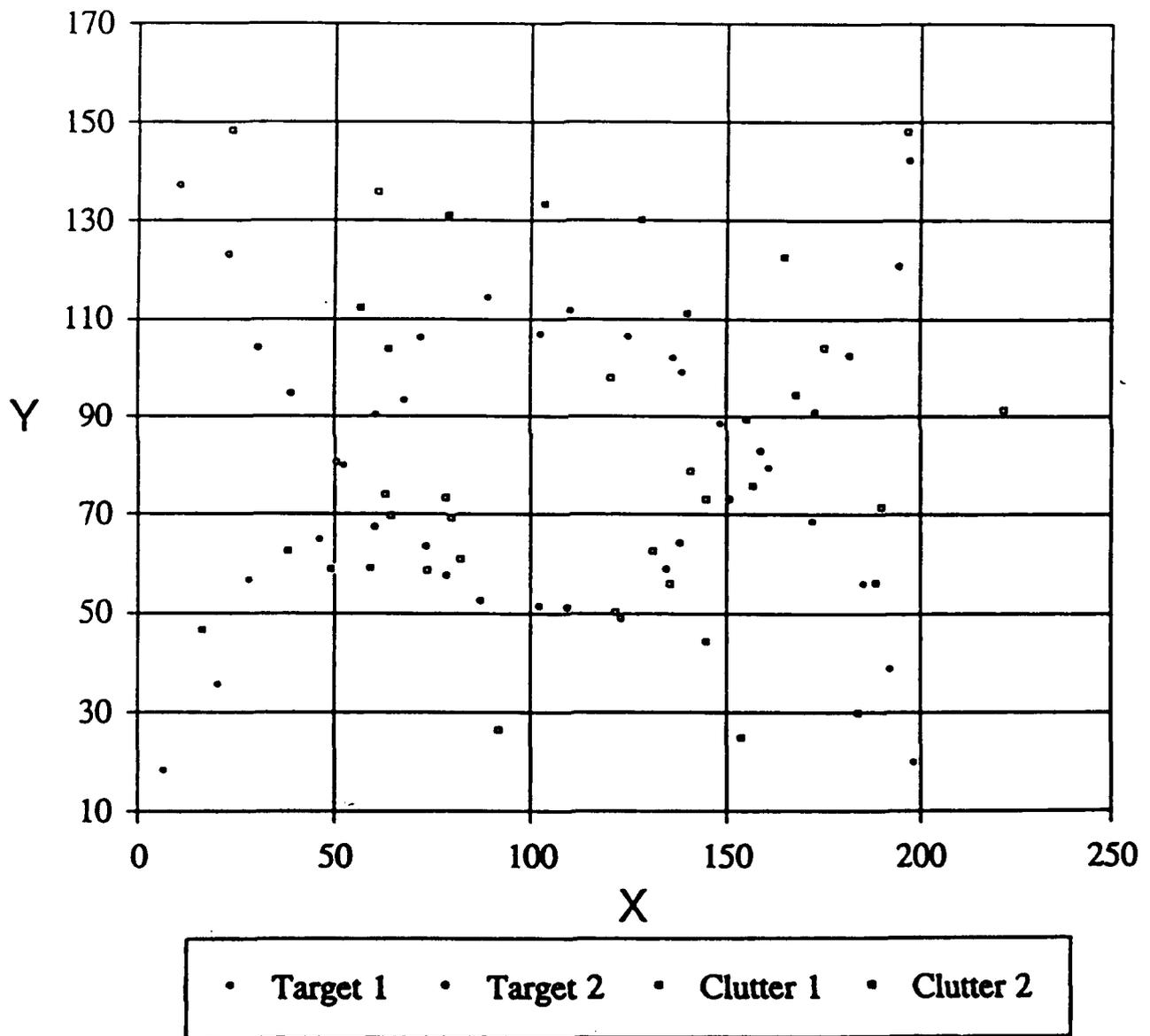
Figure 4.11: Best Three Tracks

Measurements without Noise					
Target 1	5.00	50.00	80.00	10.00	11.00
Target 2	5.00	50.00	80.00	10.00	-11.00
Measurements with Noise					
Target 1	5.00	48.92	77.50	8.38	10.20
Target 2	5.00	49.51	82.22	10.90	-11.49
Clutter Measurements					
Clutter 1	5.00	46.43	92.32	25.72	27.53
Clutter 2	5.00	50.95	66.77	15.90	-6.48
Measurements without Noise					
Target 1	6.00	60.00	90.00	10.00	9.00
Target 2	6.00	60.00	70.00	10.00	-9.00
Measurements with Noise					
Target 1	6.00	62.58	89.58	10.33	11.45
Target 2	6.00	61.25	71.06	9.69	-9.31
Clutter Measurements					
Clutter 1	6.00	76.30	81.24	0.44	1.55
Clutter 2	6.00	68.64	64.82	14.41	-4.59
Measurements without Noise					
Target 1	7.00	70.00	98.00	10.00	7.00
Target 2	7.00	70.00	62.00	10.00	-7.00
Measurements with Noise					
Target 1	7.00	71.34	97.77	9.15	8.33
Target 2	7.00	68.79	62.13	8.89	-6.55
Clutter Measurements					
Clutter 1	7.00	86.90	99.48	16.45	15.63
Clutter 2	7.00	49.69	68.01	12.33	-3.10
Measurements without Noise					
Target 1	8.00	80.00	104.00	10.00	5.00
Target 2	8.00	80.00	56.00	10.00	-5.00
Measurements with Noise					
Target 1	8.00	80.86	103.49	9.08	3.67
Target 2	8.00	79.22	52.60	9.90	-4.62
Clutter Measurements					
Clutter 1	8.00	83.37	102.06	5.64	0.23
Clutter 2	8.00	88.36	51.01	-2.40	-16.92
Measurements without Noise					
Target 1	9.00	90.00	108.00	10.00	3.00
Target 2	9.00	90.00	52.00	10.00	-3.00
Measurements with Noise					
Target 1	9.00	88.31	105.31	10.12	1.71
Target 2	9.00	89.71	53.78	9.91	-2.10
Clutter Measurements					
Clutter 1	9.00	87.66	103.61	10.25	1.84
Clutter 2	9.00	86.30	38.84	-5.24	-17.26
Measurements without Noise					
Target 1	10.00	100.00	110.00	10.00	1.00
Target 2	10.00	100.00	50.00	10.00	-1.00
Measurements with Noise					
Target 1	10.00	100.76	108.89	10.86	2.46
Target 2	10.00	100.24	51.54	8.91	-0.42
Clutter Measurements					
Clutter 1	10.00	106.42	118.92	17.58	9.18
Clutter 2	10.00	96.42	41.26	5.27	-4.06

**Table 4.2: Data for Example (continued)**

Measurements without Noise					
Target 1	17.00	170.00	68.00	10.00	-13.00
Target 2	17.00	170.00	92.00	10.00	13.00
Measurements with Noise					
Target 1	17.00	169.60	68.86	10.46	-12.75
Target 2	17.00	171.21	91.18	10.46	12.11
Clutter Measurements					
Clutter 1	17.00	171.43	80.67	-7.81	-31.02
Clutter 2	17.00	186.46	91.56	22.06	23.71
Measurements without Noise					
Target 1	18.00	180.00	54.00	10.00	-15.00
Target 2	18.00	180.00	106.00	10.00	15.00
Measurements with Noise					
Target 1	18.00	179.65	54.04	11.62	-15.83
Target 2	18.00	177.92	106.78	11.29	15.62
Clutter Measurements					
Clutter 1	18.00	189.11	64.33	15.53	-11.91
Clutter 2	18.00	169.59	109.84	3.87	8.21
Measurements without Noise					
Target 1	19.00	190.00	38.00	10.00	-17.00
Target 2	19.00	190.00	122.00	10.00	17.00
Measurements with Noise					
Target 1	19.00	190.45	38.15	9.65	-16.20
Target 2	19.00	190.34	121.77	11.12	15.48
Clutter Measurements					
Clutter 1	19.00	202.94	33.47	2.77	-23.08
Clutter 2	19.00	192.72	121.94	17.12	21.48
Measurements without Noise					
Target 1	20.00	200.00	20.00	10.00	-19.00
Target 2	20.00	200.00	140.00	10.00	19.00
Measurements with Noise					
Target 1	20.00	198.68	17.53	11.89	-16.72
Target 2	20.00	196.85	141.27	9.16	17.77
Clutter Measurements					
Clutter 1	20.00	217.71	16.06	21.97	-6.64
Clutter 2	20.00	199.96	159.62	-5.49	3.12

**Table 4.2: Data for Example (continued)**



**Figure 4.12: Noisy Target and Clutter Measurements**  
**(Clutter Radius = 50, Noise Variance = 5)**

Best two paths.

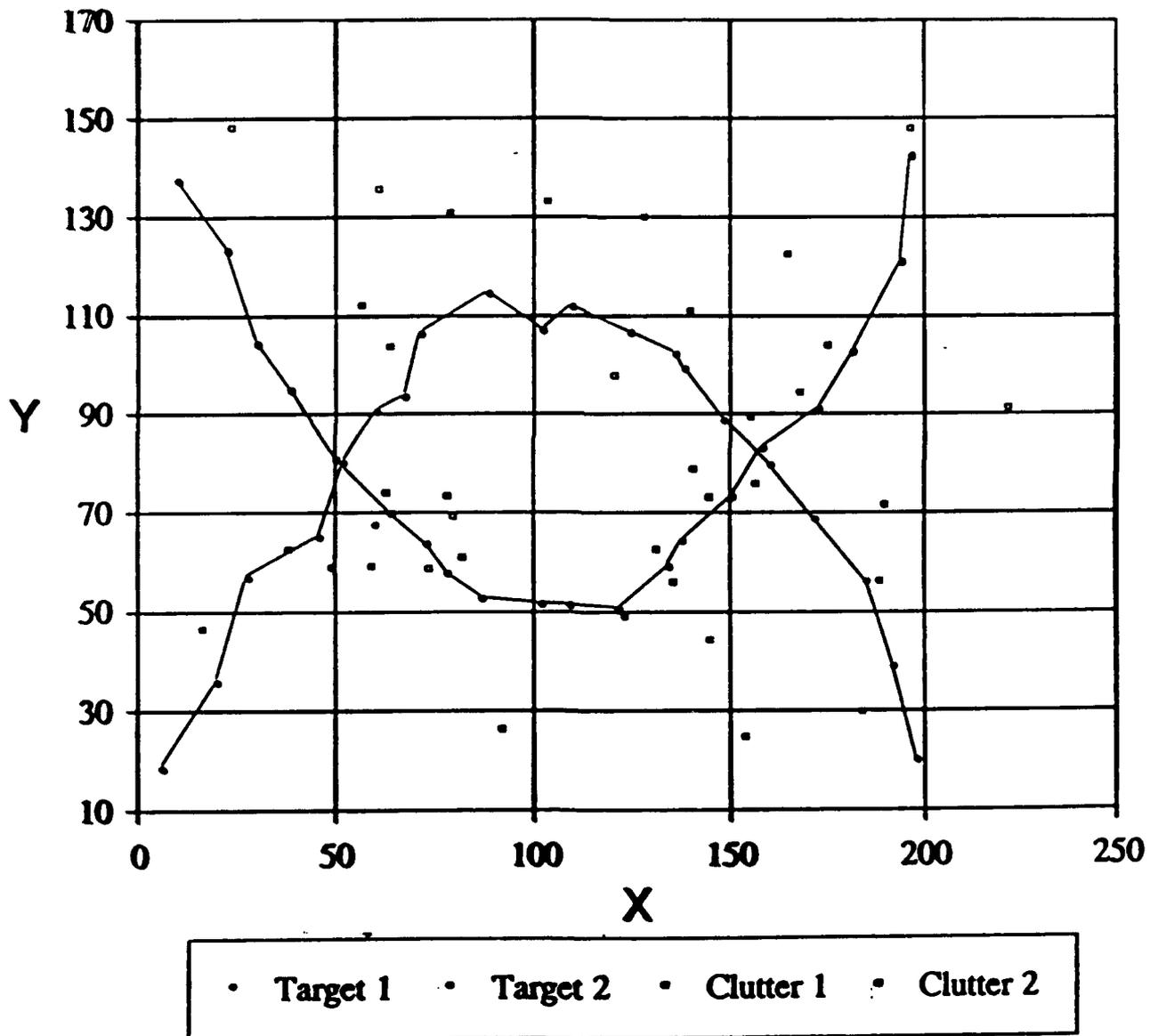


Figure 4.14: Best Two Paths

## Appendix A

### The K Best Paths Through A Trellis

We define a trellis as a type of directed graph with nodes (hereafter called states) and directed edges (hereafter called branches) that satisfy the following conditions:

1. The states are subdivided into sets, the  $i$ th set, being denoted  $T_i$ ,  $i=0, 1, 2, \dots, N$ . When we reach a state in the trellis belonging to the  $i$ th set, we say we are "at depth  $i$  in the trellis". The number of states at depth  $i$ ,  $M_i$ , can be a function of  $i$ .
2. Branches connect states at depth  $i-1$  only to states at depth  $i$ ,  $i=1, 2, \dots, N$ .

Each branch is assigned a real non-negative number called the branch metric. State metrics will also be defined in a manner to be explained shortly. A path is an ordered set of  $N$  branches which takes us from a state at depth 0 to a state at depth  $N$ . (We assume that there exists at least one path through the trellis.) The path metric is defined as the sum of the branch metrics for the  $N$  branches that make up the path.

A pair of paths will be said to be unmerged if these paths never pass through a common state. A set of paths is said to be unmerged if every pair in the set are unmerged.

It is well known that the Viterbi algorithm gives us a recursive procedure for finding the single path with the smallest path metric. Here we are concerned with finding the set of  $K$  unmerged paths ( $K \geq 1$ ) (assuming that there are at least  $K$  such paths) for which the summation of their path metrics is a minimum. We call such a set of  $K$  paths the K best paths.

So far we have made no assumptions regarding the connectivity of the trellis (i.e. the topology of the branches). Without loss of generality, we assume that every state at depth  $(i-1)$  is connected to every state at depth  $i$  for  $i=0, 1, \dots, N$ . We then take into account any missing branches by assigning the branch metric infinity to those missing branches. With this convention, the assumption that we have at least  $K$

States	Depth 0	Depth 1	Depth 2	Depth 3	Depth 4
A	0	2	2	6	3
B	0	0	2	2	4
C	0	1		3	2

Table A.1: State Metrics for Trellis of Figure A.1

As an example, consider the trellis shown in Figure A.1 for  $N=4$ . The state metrics assigned by the Viterbi Algorithm are given in Table A.1. The path with the smallest path metric is found by first noting that state C has the smallest final state metric and then tracing back the branches that gave that metric. These branches are marked with cross-hatched lines in Figure A.1 and the history of that path with the corresponding branch metrics is shown in Table A.1.

We now consider the problem of finding the  $K$  best paths through a trellis for  $K > 1$ . Before treating the problem in its most general form, let us first consider the case where the number of states at every depth in the trellis is equal to  $K$ . This situation is illustrated in Figure A.2 for the case of  $K=3$  and  $N=4$ . There are  $3!=6$  possible configurations that the best set of 3 paths can take as these paths pass from depth  $(i-1)$  to depth  $i$ ,  $i=1, 2, \dots, N$ . For example for  $i=1$ , the 6 configurations are shown in Figure A.3. In this figure we also give the sum of the branch metrics for the 3 branches that correspond to these configurations. Note that the configuration indicated in Figure A.3(a) yields the smallest sum. The sum of  $K$  path metrics,  $S$ , can be written as

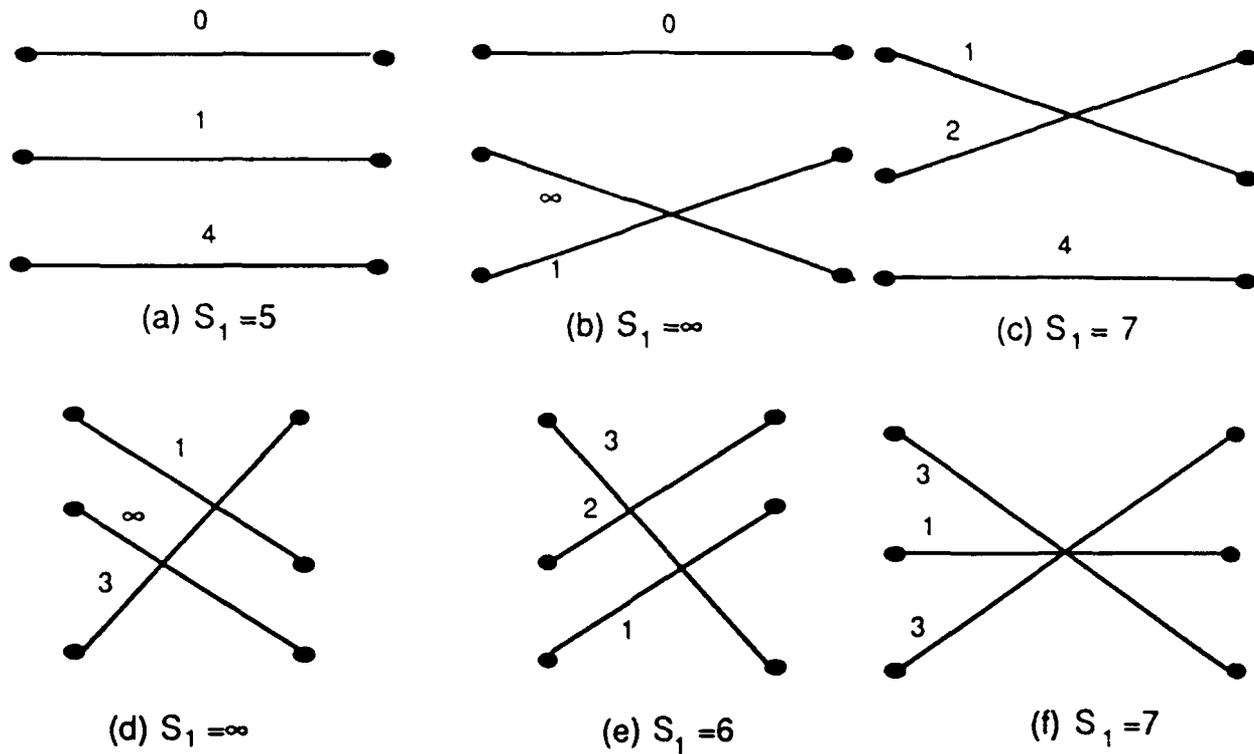


Figure A.3: Six Configurations for 3 Paths from Depth 0 to Depth 1

$\backslash i$	1	2	3	4
(a)	5	9	14	21
(b)	$\infty$	$\infty$	7	10
(c)	7	4	13	16
(d)	$\infty$	8	9	7
(e)	6	$\infty$	6	7
(f)	7	$\infty$	9	9

Table A.2: Branch Metric Sum for each Configuration

yields the best set of  $K$  paths to depth  $L$ . Said in another way, this algorithm makes a final decision on the best paths at each step of incrementing the depth.

We are now ready to consider the case of finding the best set of  $K$  paths through a trellis where the number of states at at least one depth is strictly greater than  $K$ . As an example, assume we wanted to find the two best paths through the trellis shown in Figure A.1. If these two best paths connected states  $A$  and  $B$  at depth 0 (called  $A_0, B_0$ ) to states  $A$  and  $B$  at depth 1 (called  $A_1, B_1$ ) we would know that the paths would use the best configuration that connects these states. The two possible configurations are shown in Figure A.5. Clearly the configuration shown in Figure A.5(a) is better since it leads to a smaller sum of branch metrics. Figure A.6 shows the best pair of paths that go from each pair of states at depth 0 to each pair of states at depth 1 for the trellis shown in Figure A.1.

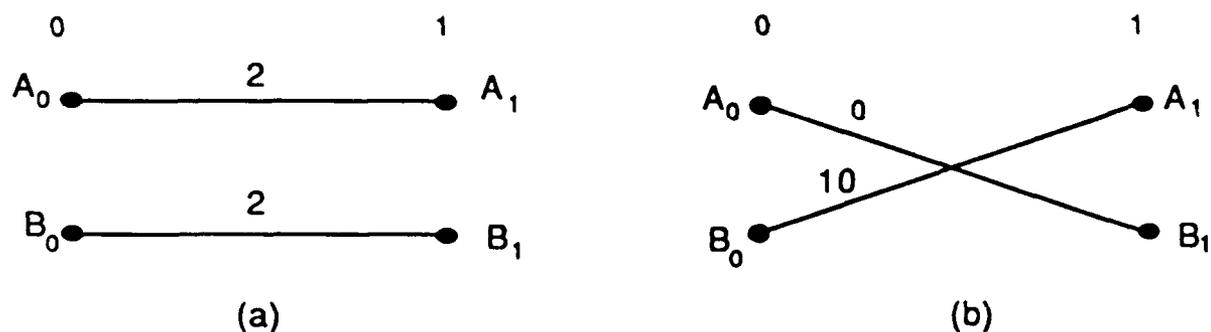


Figure A.5: Two Configurations for Connecting  $(A_0, B_0)$  to  $(A_1, B_1)$  in Figure A.1.

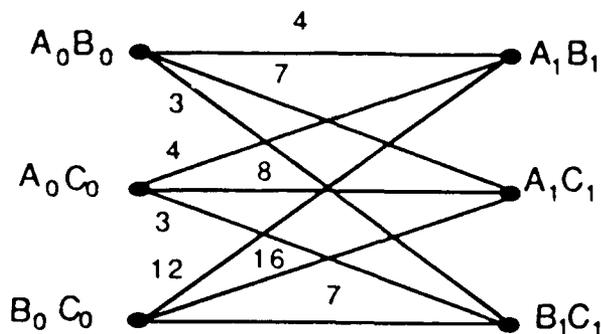


Figure A.7: One stage of Expanded Trellis for Figure A.1.

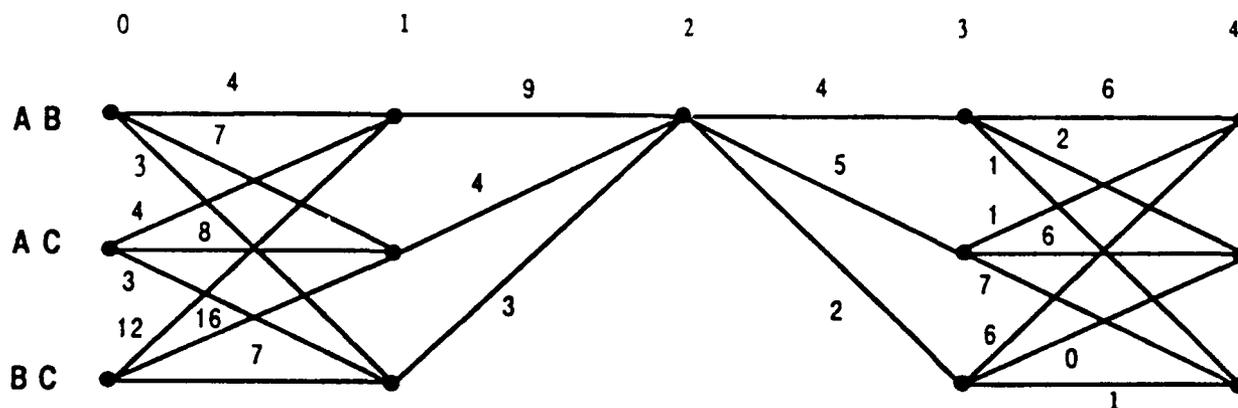


Figure A.8: Expanded Trellis for Figure A.1.

The best pair of paths through Figure A.1 is then found by using the Viterbi algorithm for the expanded trellis shown in Figure A.8. The result is summarized in Table A.3. There are two sets of 2 paths with the smallest sum metric and they are indicated by the solid lines in Table A.3 and Figure A.9. The best single path is shown by the dotted line in Figure 9. Note that the best single path is not one of the best pair of paths.

A sub-optimum algorithm can be defined which uses the Viterbi algorithm for finding the best path, deletes the states associated with that path, and then uses the Viterbi algorithm again to find the best path through the remaining states. This algorithm results in a pair of paths whose path metrics sum to 13 for the example of Figure A.1 instead of the sum of 8 corresponding to the best pair of paths.

### Algorithm 3

1. For  $i = 1, 2, \dots, N$

For  $j = 1, 2, \dots, \binom{M_{i-1}}{K}$

For  $k = 1, 2, \dots, \binom{M_i}{K}$

Of the  $K!$  possible configurations which connect the  $j$ th set of  $K$  states at depth  $i-1$  to the  $k$ th set of  $K$  states at depth  $i$ , choose the one which yields the smallest sum of branch metrics.

2. Draw an expanded trellis with  $\binom{M_i}{K}$  states at depth  $i$ , where each of these states represents a unique set of  $K$  states from the  $M_i$  states in the original trellis ( $i = 0, 1, 2, \dots, N$ ). Connect the  $j$ th state at depth  $(i-1)$  to the  $k$ th state at depth  $i$  by a branch, identifying the best of the  $K!$  configurations (as found in step 1). In particular this label includes the sum of the branch metrics for this best configuration.
3. Use the Viterbi algorithm to find the best single path through the expanded trellis. The single best path then specifies the  $K$  best paths through the original trellis.

It should be noted that Algorithm 3 reduces to Algorithm 1 for the special case of  $K=1$  (in which case the first two steps of the algorithms can be omitted) and reduces to Algorithm 2 for the special case of  $M_i=K$  for  $i=0, 1, \dots, N$ , (in which case the third step of the algorithm can be omitted).

## Appendix B

### Task Statements for Distributed Sensor Program

- Task 1                      Examine the results of prior efforts and their relation to upcoming DOD program.
- Task 2                      Examine the results of prior efforts from the point of view of communication and information theory.  
Look for performance bounds
- Task 3                      Examine pruning and merging from the point of view of coding theory, applying the techniques of sequential decoding, source coding and rate distortion theory to yield performance bounds.
- Task 4                      Examine the susceptibility to spoofing and other counter-measures.

## References

- [1] Robert A. Singer, Ronald G. Sea and Kim B. Housewright, "Derivation and Evaluation of Improved Tracking Filters for Use in Dense Multitarget Environments," IEEE Transactions on Information Theory, Vol. IT-20, No. 4, pp. 423 - 432, July 1974.
- [2] Charles L. Morefield, "Application of 0-1 Integer Programming to Multitarget Tracking Problems", IEEE Transactions on Automatic Control, Vol. AC-22, No. 3, pp. 302 - 312, June 1977.
- [3] Yaakov Bar-Shalom, "Tracking Methods in a Multitarget Environment", IEEE Transactions on Automatic Control, Vol. AC-23, No. 4, pp. 618 - 626, August 1978.
- [4] Donald B. Reid, "An Algorithm for Tracking Multiple Targets", IEEE Transactions on Automatic Control, Vol. AC-24, No. 6, pp. 843 - 854, December 1979.
- [5] Chaw-Bing Chang and John A. Tabaczynski, "Application of State Estimation to Target Tracking", IEEE Transactions on Automatic Control, Vol. AC-29, No. 2, pp. 98 - 109, February 1984.
- [6] Kuo-Chu Chang and Yaakov Bar-Shalom, "Joint Probabilistic Data Association for Multitarget Tracking with Possibly Unresolved Measurements and Maneuvers", IEEE Transactions on Automatic Control, Vol. AC-29, No. 7, pp. 585 - 594, July 1984.
- [7] Shozo Mori, Chee-Yee Chong, Edison Tse and Richard P. Wishner, "Tracking and Classifying Multiple Targets Without *A Priori* Identification", IEEE Transactions on Automatic Control, Vol. AC-31, No. 5, pp. 401 - 409, May 1986.